# KLASIFIKASI JENIS KANKER BERDASARKAN STRUKTUR PROTEIN MENGGUNAKAN METODE *NEIGHBOR WEIGHTED K-NEAREST NEIGHBOR* (NWKNN)

**SKRIPSI**

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Aldy Satria
NIM: 145150200111096

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

# PENGESAHAN

KLASIFIKASI JENIS KANKER BERDASARKAN STRUKTUR PROTEIN MENGGUNAKAN METODE *NEIGHBOR WEIGHTED K-NEAREST NEIGHBOR* (NWKNN)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Aldy Satria
NIM: 145150200111096

Skripsi ini telah diuji dan dinyatakan lulus pada
28 Desember 2018
Telah diperiksa dan disetujui oleh:

| Dosen Pembimbing I | Dosen Pembimbing 2 |
|---|---|
| Drs. Marji, M.T | Dian Eka Ratnawati , S.Si, M.Kom |
| NIP: 19670801 199203 1 001 | NIP: 19730619 200212 2 001 |

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T., Ph.D.
NIP: 19710518 200312 1 001

# PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 13 Desember 2018


Aldy Satria

NIM: 145150200111096

# PRAKATA

Assalamualaikum warahmatullahi wabarakatuh. Puji syukur keharirat Allah SWT yang telah memberikan rahmat dan karunianya hingga penulis dapat menyelesaikan laporan tugas akhir atau skripsi yang berjudul "Klasifikasi Jenis Kanker Berdasarkan Struktur Protein Menggunakan Metode Neighbor Weighted K-Nearest Neighbor (NWKNN)". Laporan tugas akhir merupakan salah satu syarat wajib untuk memperoleh gelar Sarjana Komputer (S.Kom.) dalam program studi teknik informatika. Laporan tugas akhir ini berisikan tentang implementasi klasifikasi jenis kanker berdasarkan struktur protein dengan menggunakan metode Neighbor Weighted K-Nearest Neighbor (NWKNN) yang dilakukan dengan keilmuan *data mining.* Semoga penulis maupun pembaca dapat mengambil manfaat yang sebesar-besarnya dari laporan tugas akhir ini dan ilmu yang didapat bisa dimanfaatkan sebaik-baiknya.

Laporan tugas akhir ini tidak akan tersusun dengan baik tanpa adanya dukungan dari berbagai pihak. Ucapan terimakasih tak lupa penulis sampaikan kepada:

1. Allah SWT yang selalu melimpahkan rahmat, nikmat dan kasih sayangnya kepada hambanya sehingga dapat melalui semua dengan baik.
2. Kedua orang tua penulis dan kakak serta adik-adik, Bapak Suryadi, Ibu Siti Yatimah, Asri Nova beserta keluarga kecilnya, Erlangga Darmawan dan Hafidzah Salsabila yang selalu memberikan dukungan baik moral maupun materi, semangat tiada henti dan keyakinan kepada penulis untuk dapat menyelesaikan tugas akhir ini. Terimakasih banyak atas segalanya.
3. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku dekan Fakultas Komputer Universitas Brawijaya.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku ketua jurusan Teknik Informatika yang telah berperan penting dalam pemberian izin pelaksanaan tugas akhir kepada penulis.
5. Bapak Agus Wahyu Widodo, S.T, M.Cs selaku ketua program studi Teknik Informatika yang telah berperan penting dalam proses administrasi dan izin melaksanakan tugas akhir.
6. Bapak Marji, Drs., M.T selaku dosen pembimbing I penulis yang telah membimbing penulis dalam membuat laporan tugas akhir ini.
7. Ibu Dian Eka Ratnawati , S.Si, M.Kom selaku dosen pembimbing II penulis, terimakasih atas ilmu yang telah diberikan dan kesabaran dalam mengajarkan juga pengalaman pada proses penyelesaian tugas akhir penulis.
8. Bapak Mochammad Hannats Hanafi I., S.ST., M.T dan bapak Arief Andy Soebroto, S.T, M.Kom selaku dosen pembimbing akademik yang telah memberikan saran-saran selama kegiatan perkuliahan.
9. Fauziah, Bianca dan Aca sebagai teman dekat dari bangku SMA yang selalu memberikan semangat dan support setiap harinya dan mewarnai hari-hari

penulis selama menempuh studi hingga menyelesaikan tugas akhir ini. Terima kasih sebesar-besarnya.

10. Sisca sebagai teman dekat sejak SMA, terima kasih selalu mendengarkan keluh kesah penulis, menjadi wadah curahan hati sehari-hari penulis dan juga mewarnai hari-hari penulis dengan berbagi macam cara. Terima kasih sebesar-besarnya.
11. Sadiyanti, Natallia dan Nandaini sebagai teman satu almameter di bangku SMA dan menjadi teman seperjuangan dalam menempuh studi di Malang, terima kasih atas dukungan dan semangat yang telah diberikan kepada penulis.
12. Teman-teman UKM Nol Derajat Film Universitas Brawijaya yang telah menjadi wadah diskusi dan menjadi penyemangat serta menyediakan rumah ketiga saya.
13. Teman-teman KOPMA SQUAD yang selalu mewarnai dan menemani penulis serta membantu penulis dalam pengerjaan tugas akhir ini.
14. Seluruh civitas akademik fakultas dan jurusan atas segala bantuan dalam proses administrasi dan semangat kepada penulis.

Serta pihak-pihak lain yang tidak dapat disebutkan satu-persatu oleh penulis, terimakasih atas segala bantuan sekecil apapun dan semangat yang diberikan kepada penulis selama proses pembuatan laporan tugas akhir ini sehingga dapat selesai. Sebagai manusia, penulis menyadari masih banyak kekurangan dalam laporan tugas akhir ini. Atas kekurangan tersebut penulis mohon maaf yang sebesar-besarnya. Namun, penulis berharap laporan tugas akhir ini dapat bermanfaat dan digunakan sebagaimana mestinya oleh pembaca. Terimakasih.

Malang, 13 Desember 2018

Aldy Satria

aldysatria.contact@gmail.com

# ABSTRAK

**Aldy Satria, Klasifikasi Jenis Kanker Berdasarkan Struktur Protein Menggunakan Metode Neighbor Weighted K-Nearest Neighbor (NWKNN)**

**Pembimbing: Marji, Drs., M.T dan Dian Eka Ratnawati , S.Si, M.Kom**

Kanker ialah penyakit tidak menular dengan jumlah pengidap yang besar di dunia. Kanker menjadi penyakit paling mematikan ke-7 di Indonesia. Umumnya kanker terjadi karena adanya mutasi gen yang menyebabkan adanya perubahan pada bentuk protein, salah satunya terjadi pada protein 53 (p53). Mutasi gen p53 ini sering ditemukan pada kanker manusia. Dari permasalahan ini diperlukan sebuah sistem untuk mengklasifikasikan jenis kanker. Salah satu metode yang dapat digunakan untuk klasifikasi adalah metode Neighbor Weighted K-Nearest Neighbor (NWKNN). Data yang digunakan dalam penelitian ini ialah 752 data sekuens protein dengan panjang sekuens adalah 393. Kelas klasifikasi yang digunakan berupa data bukan kanker, kanker payudara, kanker usus dan kanker paru-paru. NWKNN ialah peningkatan dari metode K-Nearest Neighbor (KNN) dengan tambahan perhitungan bobot kelas dalam perhitungan skor kelas klasifikasinya. Pengujian dilakukan dengan membagi dataset menjadi data latih dan data uji dengan varian perbandingan data latih dan data uji sebesar 90%:10%, 80%:20%, 70%:30%, 60%:40%, 50%:50%, 40%:60%, 30%:70%, 20%:80%, 10%:90% dari dataset. Hasil pengujian menunjukkan bahwa variasi perbandingan 80%:20% dengan K=8 dan E=3 menghasilkan akurasi tertinggi, yaitu 80.666%.

Kata kunci: klasifikasi, kanker, susunan protein, metode NWKNN

# ABSTRACT

**Aldy Satria, Klasifikasi Jenis Kanker Berdasarkan Struktur Protein Menggunakan Metode Neighbor Weighted K-Nearest Neighbor (NWKNN)**

**Supervisors: Marji, Drs., M.T and Dian Eka Ratnawati , S.Si, M.Kom**

Cancer is non-infectious disease with large population in the world. Cancer is ranked on 7th deadliest disease in Indonesia. Mostly cancer happened because of gene mutation that cause changes in protein form,one of them happens in protein 53 (p53). Mutation of gene p53 most commonly found in human cancers. From this case required a system that can classify the types of cancer. One of methods used is Neighbor Weighted K-Nearest Neighbor (NWKNN). Data used in this paper consists of 752 protein sequences data with 393 sequence length. Classification class includes non-cancer, breast cancer, collorectal cancer and lung cancer. NWKNN is improvement of K-Nearest Neighbor (KNN) method with addition of weight class in its classification class scoring calculation. The test is conducted by dividing dataset into training data and testing data with training data and testing data ratio 80%:20%, 70%:30%, 60%:40, 50%:50, 40%:60%, 30%:70%, 20%:80%, 10%:90% from dataset. The result shows that 80%:20% ratio with K=8 and E=3 provided the highest accuracy eate of 80.666%.

Keyword: classification, cancer, protein sequence, NWKNN method

# DAFTAR ISI

# DAFTAR TABEL

# DAFTAR GAMBAR

# BAB 1 PENDAHULUAN

## 1.1 Latar belakang

Kanker merupakan salah satu penyakit tidak menular dengan jumlah pengidap yang cukup besar. Kanker ialah suatu golongan penyakit yang muncul oleh sel tunggal yang tumbuh secara tidak normal dan tidak terkontrol. Kanker bisa terjadi di manapun, seperti pada berbagai jaringan hingga berbagai organ. Saat inilah sel-sel kanker membentuk suatu kumpulan yang masuk ke jaringan yang dekat dengan jaringan ganas dan dapat menyebar ke seluruh tubuh. Penanganan penyakit kanker biasanya dirawat dengan kemoterapi. (Mulyana, 2013).

Kanker menjadi salah satu permasalahan dalam masyarakat Indonesia. Pasalnya kanker menjadi penyakit mematikan ke-7 di Indonesia dengan persentase sekitar 5,7 % (Suryanis, 2017). Pada tahun 2013, persentase pengidap kanker di Indonesia adalah 1,4% per 100 penduduk atau bisa diperkirakan dapat mencapai 347.000 orang (Biro Komunikasi dan Pelayanan Masyarakat, 2017). Pada kasus kanker payudara, penderitanya di Indonesia juga kian meningkat dari tahun 2004 hingga tahun 2007 di mana pada tahun 2004 terdapat 5.207 kasus dan pada tahun 2007 terdapat 8.277 kasus (Anggraeni, 2017). Kemudian pada kasus kanker usus, Indonesia menjadi urutan ketiga jumlah pasiennya pada wanita dan kedua pada pria berdasarkan data GLOBOCAN 2012 (Desideria, 2017). Sedangkan pada kasus kanker paru-paru, jumlah persentase pada penduduk perempuan sebesar 13.6% (Pusat Data dan Informasi Kementerian Kesehatan RI, 2015) dan pada penduduk laki-laki sebesar 34.2% (Kementerian Kesehatan RI, 2015).

Terdapat banyak faktor pemicu kanker sehingga terdapat pula berbagai macam jenis kanker, namun pada umumnya kanker disebabkan oleh adanya mutasi gen, salah satunya ialah p53. Gen p53 terjadi apabila adanya mutasi DNA. Mutasi tersebut menyebabkan adanya perbedaan urutan susunan asam amino protein p53 (Kurnianti, 2013). Mutasi p53 dapat menghilangkan fungsi *p53 wild type* (normal) yang berfungsi untuk mengendalikan siklus sel (Retwitasari, 2016).

Bioinformatika menjadi salah satu topik yang hangat diperbincangkan karena topik ini terus dikembangkan belakangan ini. Penggunaan data yang digunakan ialah data yang berkaitan dengan bidang kedokteran dan biologi yang dikomputasi oleh sistem komputasi. Contohnya data-data ini digunakan dalam mendeteksi suatu penyakit. Salah satu data yang digunakan dalam bidang ini ialah data protein.

Data yang diolah pada bioinformatika biasanya berskala besar dan jenisnya dapat bermacam-macam. Pemrosesan pencarian informasi yang dibutuhkan juga menjadi lebih lama dan juga kurang akurat pada suatu kasus sehingga diperlukan data mining dalam mengatasinya.

Data mining didefinisikan sebagai proses menemukan atau proses pengesktrakan informasi baru yang berguna di mana data tersebut berasal dari kumpulan basis data yang besar untuk pengambilan sebuah keputusan (Prasetyo,

2012). Penggunaan data mining ini diharapkan bisa mengenali kumpulan pola dalam gudang basis data dengan input yang minimal pada sistem (Hermawati, 2013). Data mining bertujuan untuk mengklasifikasi sebuah data yang diinput ke dalam jenis/golongan yang telah terdefinisi sebelumnya.

Terdapat banyak metode klasifikasi yang dapat digunakan dalam data mining seperti *K-Nearest Neighbor, Naive Bayes Classifier, Artificial Neural Network* (jaringan syaraf tiruan), *Support Vectro Maching (SVM), Fuzzy K-Nearest Neighbor*. Metode *K-Nearest Neighbor* merupakan salah satu metode data mining yang sering digunakan dikarenakan metode ini cukup mudah dalam implementasinya. Metode K-NN ini melakukan proses klasifikasi dengan cara mencari kedekatan lokasi atau jarak suatu data dengan data lain yang berada pada kumpulan basis data yang telah terdefinisi kelasnya (Prasetyo, 2012). Penelitian sebelumnya yang dilakukan oleh Ria Kurniati (2013) untuk klasifikasi jenis kanker berdasarkan susunan protein juga menerapkan metode pengelompokan K-Means pada klasifikasi K-NN. Dalam penelitian ini pengelompokan data latih yang dilakukan sebelum proses klasifikasi memiliki tujuan agar hasil klasifikasi dapat lebih optimal. Namun, peneliti menyatakan bahwa metode K-Means bergantung pada penentuan *centroid* awal dan menyarankan menggabungkan metode lain agar dapat lebih optimal. Penelitian lain dengan kasus yang sama juga dilakukan oleh Arintha Retwintasari (2016) menggunakan metode *Modified K-Nearest Neighbor (MKNN)* dan menghasilkan nilai akurasi maksimum sebesar 45.53% dengan dataset sebanyak 150 data dan nilai K berjumlah 1.

Penelitian sebelumnya dilakukan oleh Fadila (2016) terhadap implementasi metode klasifikasi *Neighbor Weighted K-Nearest Neighbor (NWKNN)* untuk identifikasi jenis *Attention Deficit Hyperactivity Disorder (ADHD*) pada anak usia dini menunjukkan bahwa rata-rata akurasi yang dihasilkan ialah 78%. Berdasarkan latar belakang yang sudah dipaparkan, maka judul skripsi ini adalah "KLASIFIKASI JENIS KANKER BERDASARKAN STRUKTUR PROTEIN MENGGUNAKAN METODE *NEIGHBOR WEIGHTED K-NEAREST NEIGHBOR (NWKNN)*".

## 1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dipaparkan, maka rumusan masalah yang bisa dikaji di skripsi ini adalah:

1. Bagaimana menerapkan metode *Neighbor Weighted K-Nearest Neighbor (NWKNN)* untuk klasifikasi jenis penyakit kanker?
2. Bagaimana tingkat akurasi hasil klasifikasi jenis penyakit kanker dengan menerapkan metode *Neighbor Weighted K-Nearest Neighbor (NWKNN)*?

## 1.3 Tujuan

Berdasarkan dari rumusan masalah tersebut, tujuan dalam penelitian ini ialah:

1. Untuk menerapkan metode *Neighbor Weighted K-Nearest Neighbor* pada *sequence* p53.

2. Untuk mengetahui tingkat akurasi dalam memprediksi jenis penyakit kanker menggunakan metode *Neighbor Weighted K-Nearest Neighbor* (NWKNN).

## 1.4 Manfaat

Manfaat yang dapat diperoleh bagi penulis adalah memperoleh pengetahuan dan dapat lebih memahami metode *Neighbor Weighted K-Nearest Neighbor* (NWKNN) untuk klasifikasi jenis penyakit kanker. Selain itu dapat memudahkan dalam mengetahui apakah suatu kasus terkena jenis kanker payudara, usus, paru-paru atau tidak kanker dengan berdasarkan struktur protein.

## 1.5 Batasan masalah

Masalah pada skripsi ini dibatasi pada hal-hal berikut:

1. Algoritma klasifikasi hanya menggunakan *Neighbor Weighted K-Nearest Neighbor (NWKNN)*.
2. Objek yang digunakan untuk dasar klasifikasi penyakit kanker ialah data susunan protein dalam tubuh manusia.
3. Klasifikasi jenis kanker yang digunakan ialah jenis kanker paru-paru, kanker kolon (usus) dan kanker payudara.
4. Data yang digunakan dalam melakukan pengujian metode didapat dari website http://uniprot.org dan data susunan protein yang diambil memiliki panjang yang sama.

## 1.6 Sistematika pembahasan

Demi mencapai tujuan yang diharapkan, skripsi ini disusun berdasarkan sistematika penulisan sebagai berikut:

1. **BAB 1 PENDAHULUAN**

   Bab ini membahas tentang latar belakang penulisan skripsi ini, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika pembahasan.

2. **BAB 2 LANDASAN KEPUSTAKAAN**

   Bab ini berisi tentang penelitian terkait yang telah dilakukan. Beberapa dasar teori yang diperlukan untuk menunjang penelitian ini adalah teori penyakit kanker, jenis-jenis penyakit kanker, *data mining*, dan metode *Imporved K-Nearest Neighbor (NWKNN).*

3. **BAB 3 METODOLOGI**

   Bab ini berisi tentang tahapan yang akan diambil untuk penentuan jenis kanker berdasarkan sturktur protein menggunakan metode *Neighbor Weighted K-Nearest Neihbor* (NWKNN).

4. **BAB 4 PERANCANGAN**

Bab ini berisi tentang implementasi (perangkat keras, sistem operasi, bahasa pemrograman yang digunakan), batasan-batasan implementasi serta algoritma operasi-operasi yang akan diimplementasikan untuk sistem klasifikasi jenis penyakit kanker berdasarkan struktur protein.

5. **BAB 5 IMPLEMENTASI**

   Bab ini berisi tentang penjelasan/pembahasan dari implementasi klasifikasi jenis kanker berdasarkan struktur protein menggunakan metode *Neighbor Weighted K-Nearest Neighbor (NWKNN)*.

6. **BAB 6 PENGUJIAN DAN ANALISIS**

   Bab ini berisi hasil pengujian yang dilakukan pada klasifikasi jenis kanker berdasarkan struktur protein menggunakan Neighbor Weighted K-Nearest Neighbor (NWKNN) serta analisis terhadap hasil pengujian yang telah dilakukan.

7. **BAB 7 PENUTUP**

   Bab ini berisi kesimpulan yang didapat dari hasil pengujian serta saran-saran untuk pengembangan lebih lanjut.

# BAB 2 LANDASAN KEPUSTAKAAN

## 2.1 Kajian Pustaka

Penelitian ini membahas tentang pengklasifikasian jenis kanker berdasarkan susunan protein dengan menggunakan metode *Neighbor Weighted K-Nearest Neighbor (NWKNN)*. Sebelumnya telah dilakukan penelitian lain oleh Putri Nur Fadila (2016) identifikasi jenis *Attention Deficit Hyperactivity Disorder (ADHD)* pada anak usia dini menggunakan metode *Neighbor Weighted K-Nearest Neightbor (NWKNN)*. Hasil percobaan menunjukkan bahwa sistem yang diimplementasikan memiliki kinerja sistem yang mampu berjalan sesuai dengan kebutuhan fungsionalnya dan memiliki rata-rata akurasi yang cukup tinggi, yaitu hingga sebesar 78%. Selain itu, NWKNN juga menghasilkan rata-rata akurasi 2% lebih baik dibanding dengan KNN.

Metode NWKNN juga diterapkan pada klasifikasi penyimpangan tumbuh kembang pada anak pada penelitian yang dilakukan oleh Afrizal (2017). Hasil penelitian tersebut menunjukkan bahwa hasil akurasi terbaik didapatkan pada saat menggunakan rasio data latih dan data uji 80:20 dengan nilai K=10 dan nilai E=4 dengan mencapai akurasi sebesar 95%. Pada percobaan perbandingan akurasi metode NWKNN dengan KNN, didapatkan bahwa hasil akurasi metode NWKNN 5-15% lebih baik dibanding dengan metode KNN setelah nilai K semakin meningkat (setelah K=2). Hal ini dikarenakan adanya proses pembobotan pada setiap kelas yang dapat membantu mengenali kelas yang berasal dari data yang merupakan minoritas.

Implementasi *Fuzzy K-Nearest Neighbor* untuk klasifikasi jenis kanker berdasarkan susunan protein dilakukan oleh Tahtri Nadia Utami (2018). Pada penelitian dengan metode *Fuzzy K-Nearest Neighbor* untuk klasifikasi kanker ini menunjukkan bahwa nilai K yang terbaik adalah nilai K sebesar 5 dengan rata-rata tingkat akurasi sebesar 54.99%. Kemudian rata-rata akurasi terbesar untuk pengujian *k-fold-validation* sebesar 52.56%. Pengujian lain yang dilakukan ialah pengujian variasi jumlah data latih yang besar dengan jumlah data latih sebesar 90% dari dataset menghasilkan akurasi tertinggi sebesar 55.33%.

Penelitian terkait tentang penentuan jenis kanker berdasarkan susunan protein telah dilakukan oleh Ria Kurnianti (2013) dengan menggunakan metode pengelompokkan K-Means. Penelitian tersebut menunjukkan bahwa penelitian tersebut memiliki nilai K yang optimal pada proses pengelompokkan ketika nilai T=15 dengan K=12 karena menghasilkan *error rate* sebesar 6.16% dengan rata-rata K optimal didapatkan pada saat T=10 dengan K=6. Peneliti juga menyimpulkan bahwa data yang semakin menyebar mengakibatkan nilai *error rate* yang semakin tinggi.

Penelitian lain tentang penentuan jenis kanker berdasarkan susunan protein juga dilakukan oleh Arintha Retwitasari (2016). Peneliti menggunakan algoritma *Modified K-Nearest Neighbor (MKNN)* dalam penentuan jenis kanker. Pengujian sistem dilakukan dengan menggunakan 3 jumlah dataset yang berbeda, masing-

masing dataset berjumlah 100 data, 150 data, dan 200 data. Pada masing-masing dataset memiliki persentase data latih : data uji yang berbeda, yaitu 80%:20%, 70:30%, dan 60%:40%. Pengujian ini dilakukan untuk mengetahui pengaruh jumlah data uji terhadap nilai akurasi, mengetahui pengaruh nilai K terhadap tingkat akurasi, dan pengaruh jumlah data latih terhadap tingkat akurasi. Penelitian tersebut menunjukkan bahwa rata-rata nilai akurasi maksimum yang dihasilkan sistem sebesar 43,53% ketika dataset berjumlah 150 data dan akurasi minimum sebesar 37,93% ketika dataset berjumlah 200 data. Kesimpulan lain ialah semakin tinggi persentase data uji belum sepenuhnya benar membuat tingkat akurasi semakin tinggi, rata-rata nilai akurasi cenderung menurun ketika nilai K juga bertambah dan banyaknya jumlah dataset juga mempengaruhi tingkat akurasi.

Penelitian dengan metode K-Medoids untuk clustering pasien kanker berdasarkan struktur protein dalam tubuh yang dilakukan oleh Laily R. Putri Rizby (2018) dengan metode K-Medoids menunjukkan bahwa didapat hasil terbaik pada pengujian pengaruh jumlah cluster yaitu dengan cluster = 14 yang memiliki *silhouette coefficient* 0,726 dan jumlah dataset 116 data atau 20% memiliki nilai *silhouette coefficient* 0,778. Selain itu nilai *silhouette coefficient* selalu berada di atas 0,5 yang telah diketahui sebelumnya jika nilai *silhouette coefficient* berada pada nilai -1 maka data berada pada cluster yang salah dan jika nilai *silhouette coefficient* mendekati 1, maka data berada pada cluster yang tepat.

Penelitian mengenai klasifikasi jenis kanker berdasarkan struktur protein juga dilakukan oleh Tawang Wulandari (2018) dengan menggunakan *Naive Bayes*. Penelitian tersebut menyebutkan bahwa tingkat akurasi paling kecil yang dihasilkan sistem ialah 25,00% pada pengujian 588 dataset pada persentase uji 10% dari seluruh dataset, sedangkan nilai akurasi terbesar yang dihasilkan sistem adalah 79,17% pada pengujian 848 dataset pada persentase data uji 60,00%. Rata-rata akurasi terbesar yang dihasilkan sistem adalah 67,16% pada pengujian menggunakan 848 dataset.

## 2.2 Dasar Teori

### 2.2.1 Kanker

Kanker menjadi penyakit kedua yang menyebabkan kematian terbesar di dunia. Kanker berkembang ketika sel normal di beberapa titik tertentu mulai tumbuh tak terkontrol. Ada banyak tipe kanker, namun semua tipe sel kanker terus tumbuh, membelah dan membelah lagi dibanding mati dan membentuk sel abnormal baru. Beberapa tipe sel kanker sering berpindah ke titik tubuh yang lain melalui sirkulasi darah. Secara umum sel kanker berkembang dari sel normal dikarenakan DNA yang rusak. Biasanya ketika DNA rusak, tubuh dapat memperbaikinya, sayangnya pada kasus sel kanker, DNA yang rusak tidak dapat diperbaiki. Manusia bisa juga menuruni DNA rusaknya dari orang tuanya (Sudhakar, 2009).

#### *2.2.1.1 Kanker Payudara (Breast Cancer)*

Kanker payudara ialah masalah besar di Indonesia. Wanita menjadi pengidap mayoritas pada penyakit ini, tapi tidak menutup kemungkinan terjadi pada laki-laki.

Pada jurnal kesehatan masyarakat yang dilakukan oleh Lindra Anggorowati (2013) menyebutkan bahwa obesitas ialah faktor risiko terjadinya kanker payudara karena obesitas akan meningkatkan sistesis estrogen yang akan menimbulkan dampak terhadap proses proliterasi jaringan payudara karena timbunan lemak. Faktor lain yang menjadi timbulnya penyakit kanker payudara ialah faktor usia, usia *menarche*, usia menopause, lama menyusui, lama pemakaian kontrasepsi, pola konsumsi makanan berlemak, aktivitas fisik (Yulianti, et al., 2016).

#### *2.2.1.2 Kanker Usus (Colorectal Cancer)*

Kanker usus terjadi karena *adenokarsinoma* yang tumbuh dari *polypadenoma*. Kanker ini pertumbuhannya tidak terdeteksi dengan mudah walaupun menimbulkan beberapa gejala. Gejala awal yang bisa terjadi seperti terjadinya perubahan kebiasaan buang air besar, diare hingga konstipasi. Pada saat terjadinya gejala ini, tumor bisa jadi telah menyebar ke dalam lapisan yang lebih dalam ke organ-organ yang berdekatan, termasuk jaringan usus. Perluasan penyebaran kanker ini berlangsung ke sekeliling permukaan usus, submukosa dan dinding luar usus (Harahap, 2004).

#### *2.2.1.3 Kanker Paru-Paru (Lung Cancer)*

Kanker paru-paru ialah suatu kondisi di mana tumor ganas menyerang pada saluran pernapasan dan paru, yaitu dari saluran bronkus dan sel alveoli di dalam paru-paru. Selain itu kanker paru juga bisa terjadi karena adanya penyebaran dari organ yang lain, seperti penyebaran dari ginjal, lambung, payudara, usus besar, leher rahim, rektum, tulang, buah zakar, kulit dan prostat. Sebagian besar kanker paru dapat didiagnosa ketika sang penderita berada di stadium lanjut yang menyebabkan buruknya prognosis. Faktor utama terjadinya kanker ini ialah ditemukannya kebiasaan merokok pada pengidapnya (Viswasnathan, 2016).

### 2.2.2 Protein

Protein merupakan makromolekul yang terdiri dari bahan dasar asam amino, di mana terdapat 20 macam asam amino yang menyusun protein. Protein memiliki peran yang kompleks dalam proses biologi. Protein berperan menjadi katalisator, yaitu menjadi penyimpan dan penghantar molekul lain, mendukung sistem imunitas tubuh, menghasilkan pergerakan tubuh, sebagai transmitor gerakan syaraf dan mengontrol dan perkembangan (Katili, 2009).

Terdapat empat bentuk struktur asam amino, yaitu struktur primer, struktur sekunder, struktur tersier dan struktur kuartener (Sari, 2007).

1. Struktur Primer: adalah urutan asam amino yang tersusun secara linear yang digabungkan dengan ikatan peptida yang mencakup lokasi setiap ikatan disulfida.
2. Struktur Sekunder: adalah daerah yang berada di dalam rantai peptida yang membentuk struktur reguler, berulang, dan lokal yang terbentuk karena adanya ikatan hidrogen antara atom-atom ikatan peptida yang mempengaruhi posisi kedekatan antara ruang residu asam amino dengan barisan linear.
3. Struktur Tersier: adalah hubungan spasial antar unsur struktur sekunder.
4. Struktur Kuartener: menggambarkan pengaturan subunit protein dalam ruang. Pada protein ini, setiap rantai polipeptida disebut subunit di mana subunit ini digabungkan dengan jenis interaksi nonkovalen yang berperan dalam struktur tersier.

#### *2.2.2.1 Kode Genetik Terhadap Protein*

Setidaknya ada paling sedikit 3 residu nukleotida DNA yang diperlukan untuk mengode masing-masing asam amino sejak tahun 1960. Empat huruf kode DNA yaitu A, T, G dan C tersusun dan membentuk 3 huruf yang disebut sebagai kodon. Ketika terjadi proses di dalam sel, proses transkripsi juga terjadi. Proses transkripsi merupakan sintesis RNA dan DNA dijadikan untuk cetakannya. RNA yang membawa pesan yang sama dengan resep pada DNA ini yang bertindak sebagai cetakan untuk sintesis protein. Masing-masing kodon mengodekan 1 asam amino. Di lain sisi, jumlah asam amino penyusun protein diketahui sebanyak 20 (dengan beberapa tambahan asam amino yang jarang). Dengan demikian berarti ada asam amino yang dikodekan oleh lebih dari satu kodon (Retwitasari, 2016).

### 2.2.3 Mutasi

Mutasi ialah suatu kondisi di mana materi genetik pada suatu makhluk terjadi perubahan secara tiba-tiba. Mutasi gen yaitu perubahan terjadi pada material genetik. Mutasi terjadi akibat adanya perubahan urutan (*sequence*) nukleotida DNA kromosom yang mengakibatkan terjadinya perubahan pada bentuk protein. Mutasi pada sekuens DNA gen bisa mengubah bentuk urutan asam amino dari protein yang dikode oleh gen (Retwitasari, 2016).

### 2.2.4 Bioinformatika

Bioinformatika dikenalkan setelah para biologis menemukan cara mengurutkan DNA dan membuat banyak teks dengan empat alfabet pada DNA. Cara yang digunakan para ahli bioinformatika untuk mengurai bahasa DNA adalah dengan algoritma, statistik dan teknik matematika lainnya (Jones & Pevzner, 2004). Bioinformatika ditujukan untuk menjawab masalah biologi dengan menggunakan sekuen DNA dan asam amino dan informasi lain yang terkait (Retwitasari, 2016).

#### *2.2.4.1 Protein Sequencing*

*Protein sequencing* merupakan penentuan proses penentuan urutan asam amino di suatu protein. Hal ini sangat bermanfaat untuk menemukan gen sejenis

di beberapa organisme bahkan untuk mengecek apakah hasil sekuensinya telah absah (Retwitasari, 2016).

#### *2.2.4.2 Subtitution Matrix*

Matriks subtitusi (*subtitution matrix*) ialah suatu matriks kesamaan (*similarity matrix*) yang berfungsi untuk deklarasi *residue substitution score*. Contohnya adalah PAM (*Point Accepted Mutation Matrix*).

#### *2.2.4.3 PAM (Point Accepted Mutation)*

PAM (*Point Accepted Mutation*) merupakan kumpulan PAM1 – PAM250 yang berasal dari turunan *sequence* yang memiliki relasi kekerabatan yang dekat. Terdapat dua proses yang berbeda saat terjadi perubahan pada asam amino, yaitu:

1. Mutasi terjadi pada bagian gen yang memproduksi asam amino dari protein.
2. Mutasi terjadi oleh jenis baru yang lebih dominan. Asam amino baru seringkali membuat dirinya menyerupai dan memiliki fungsi dengan asam amino yang lama agar asam amino yang baru bisa diterima oleh asam amino yang lain.

Matriks PAM1 adalah dasar untuk menghitung matriks yang lain dengan anggapan mutasi yang berulang akan mengikuti aturan yang sama dengan matriks PAM1, dengan logika tersebut dapat diperoleh matriks PAM250 (Kurnianti, 2013). Tabel 2.1 menunjukkan tabel matriks PAM250.

**Tabel 2.1 Matriks PAM250**

| | A | R | N | D | C | Q | E | G | H | I | L | K | M | F | P | S | T | W | Y | V |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 2 | -2 | 0 | 0 | -2 | 0 | 0 | 1 | -1 | 0 | -2 | -1 | -1 | -3 | 1 | 1 | 1 | -6 | -4 | 0 |
| R | -2 | 6 | 0 | -1 | -4 | 1 | -1 | -3 | 2 | -2 | -3 | 3 | 0 | -4 | 0 | 0 | -1 | 2 | -5 | -2 |
| N | 0 | 0 | 2 | 2 | -3 | 1 | 2 | 1 | 2 | -2 | -3 | 1 | -2 | -3 | 0 | 1 | 1 | -4 | -2 | -2 |
| D | 0 | -1 | 2 | 4 | -5 | 2 | 3 | 1 | 1 | -2 | -4 | 0 | -3 | -5 | -1 | 0 | 0 | -7 | -4 | -2 |
| C | -2 | -3 | -4 | -5 | 12 | -5 | -5 | -3 | -3 | -2 | -6 | -5 | -5 | -4 | -3 | 0 | -2 | -8 | 0 | -2 |
| Q | 0 | 1 | 1 | 2 | -5 | 4 | 2 | -1 | 3 | -2 | -2 | 1 | -1 | -4 | 0 | -1 | -1 | -5 | -4 | -2 |
| E | 0 | -1 | 1 | 3 | -5 | 2 | 4 | 0 | 1 | -2 | -3 | 0 | -2 | -5 | 0 | 0 | 0 | -7 | -4 | -2 |
| G | 1 | -3 | 0 | 1 | -3 | -1 | 0 | 5 | -2 | -3 | -4 | -2 | -3 | -5 | 0 | 1 | 0 | -7 | -5 | -1 |
| H | -1 | 1 | 1 | 1 | -3 | 3 | 0 | -2 | 6 | -2 | -2 | 0 | -2 | -2 | 0 | -1 | -1 | -3 | 0 | -2 |
| I | -1 | -2 | -2 | -2 | -2 | -2 | -2 | -3 | -2 | 5 | 2 | -2 | 2 | 1 | -2 | -1 | 0 | -5 | -1 | 4 |
| L | -2 | -3 | -3 | -4 | -6 | -2 | -3 | -4 | -2 | 2 | 6 | -3 | 4 | 2 | -3 | -3 | -2 | -2 | -1 | 2 |
| K | -1 | 3 | 1 | 0 | -5 | 1 | 0 | -2 | 0 | -2 | -3 | 5 | 0 | -5 | -1 | 0 | 0 | -4 | -5 | -2 |
| M | -1 | -1 | -2 | -3 | -5 | -1 | -2 | -3 | -2 | 2 | 4 | 1 | 6 | 0 | -2 | -2 | 0 | -4 | -3 | 2 |
| F | -3 | -4 | -3 | -5 | -4 | -4 | -5 | -4 | -2 | 1 | 2 | -5 | 0 | 9 | -5 | -3 | -3 | 0 | 7 | -1 |
| P | 1 | 0 | 0 | -1 | -3 | 0 | 0 | 0 | 0 | -2 | -3 | -1 | -2 | -4 | 6 | 1 | 0 | -6 | -5 | -1 |
| S | 2 | 1 | 2 | 1 | 1 | 0 | 1 | 2 | 0 | -1 | -2 | 1 | -1 | -2 | 2 | 2 | 2 | -2 | -2 | 0 |
| T | 0 | -2 | 0 | -1 | -3 | -2 | -1 | -1 | -2 | -1 | -2 | -1 | -1 | -4 | 0 | 1 | 2 | -6 | -4 | 0 |
| W | -6 | 2 | -5 | -7 | -7 | -6 | -7 | -7 | -5 | -6 | -7 | -4 | -6 | 1 | -6 | -2 | -5 | 17 | 1 | -8 |
| Y | -3 | -5 | -2 | -4 | 1 | -4 | -4 | -5 | 0 | -1 | -1 | -5 | -2 | 7 | -5 | -3 | -3 | 0 | 10 | -2 |
| V | 0 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | 4 | 2 | -2 | 2 | -1 | -1 | -1 | 0 | -6 | -3 | 4 |

### 2.2.5 Data Mining

Data mining didefinisikan sebagai proses menemukan atau proses pengesktrakan informasi baru yang berguna di mana data tersebut berasal dari kumpulan basis data yang besar untuk pengambilan sebuah keputusan. (Prasetyo, 2012). Data mining juga disebut pembelajaran berbasis induksi di mana proses pencarian definisi dilakukan dengan cara mencari contoh-contoh spesifik yang

akan dipelajari secara otomatis. Penggunaan data mining ini diharapkan bisa mengenali kumpulan pola dalam gudang basis data dengan input yang minimal pada sistem (Hermawati, 2013). Data mining bertujuan untuk mengklasifikasi sebuah data yang diinput ke dalam jenis/golongan yang telah terdefinisi sebelumnya.

Sifat data mining dapat dibedakan menjadi dua sifat, yaitu prediksi (*prediction driven*) yang digunakan untuk menjawab suatu pertanyaan dan sesuatu yang sifatnya masih belum jelas, dan yang kedua ialah penemuan (*discovery driven*) yang biasanya untuk menjawab pertanyaan sebab-akibat dan punya sifat transparan (Hermawati, 2013).

Data mining kerap dianggap sebagai satu langkah proses dari *Knowledge Discovery in Databases (KDD)*, yaitu sebuah penerapan metode sains pada data mining. Tahapan proses penggunaan data mining yang merupakan proses KDD yaitu (Hermawati, 2013):

1. Tahap mengetahui dan menggali pengetahuan awal dan sasaran pengguna dari memahami domain aplikasi.
2. Membuat target dataset yang mencakup pemilihan data dan fokus pada sub-set data.
3. Melakukan pembersihan dan transformasi data serta pemilihan fitur dan reduksi dimensi.
4. Pada tahap ini data mining digunakan yang terdiri dari asosiasi, sekuensial, klasifikasi, klasterisasi, dll.
5. Interpretasi, evaluasi dan visualisasi pola yang berguna untuk mengecek apabila terdapat sesuatu yang baru dan dilakukan iterasi jika dilakukan.

#### *2.2.5.1 Klasifikasi*

Klasifikasi merupakan sebuah proses mendapatkan objek data untuk didefinisikan ke dalam kelas tertentu dari beberapa kelas berbeda yang tersedia. Kalsifikasi juga disebut sebagai suatu proses pembelajaran untuk dilakukan pelatihan terhadap suatu fungsi yang mengalokasikan setiap fitur ke salah satu kelas yang tersedia di mana proses ini disimpan sebagai memori (Prasetyo, 2012).

Terdapat banyak metode klasifikasi yang dapat digunakan dalam data mining seperti K-Nearest Neighbor, Naive Bayes Classifier, *Artificial Neural Network* (jaringan syaraf tiruan), *Support Vectro Maching (SVM)*, *Fuzzy K-Nearest Neighbor*.

#### *2.2.5.2 K-Nearest Neighbot (K-NN)*

*K-Nearest Neighbor (K-NN)* merupakan suatu metode klasifikasi yang menentukan obyek latih ke kelas yang mempunyai sifat ketetanggaan (*neighborhood*) yang paling dekat (Meristika, 2013).

#### *2.2.5.3 Proses K-Nearest Neighbor (K-NN)*

Menurut Fadila (2016) langkah pada metode KNN meliputi:

a. Menentukan nilai parameter K.

b. Menghitung nilai sifat ketetanggaan antara data uji terhadap data latih menggunakan *Euclidean Distance* ataupun *Cosine Similarity* (*CosSim*). Namun pada penelitian ini menggunakan *Cosine Similarity* untuk menghitung kedekatan ketetanggaannya. Penggunaan CosSim untuk menghitung kedekatan ketetanggaannya digunakan karena merujuk oleh penelitian sebelumnya yang dilakukan oleh Tan (2005). Rumus penghitungan *CosSim* ditunjukkan pada Persamaan 2.1.

$$\text{CosSim}(q, d_j) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|} = \frac{\sum_{i=1}^{m}(w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{m} {w_{ij}}^2 \cdot \sum_{i=1}^{m} {w_{iq}}^2}} \quad (2.1)$$

Di mana:

$\text{CosSim}(q, d_j)$ = nilai *CosSim*, similaritas data uji x dengan data latih i

$\vec{q}$ = data uji

$\vec{d}_j$ = data latih

$\vec{d}_j \cdot \vec{q}$ = hasil total perkalian vektor antara data latih dengan data uji

$|\vec{d}_j| \cdot |\vec{q}|$ = hasil total perkalian vektor antara norm data latih dengan data uji

$w_{ij}$ = bobot nilai i pada data latih j

$w_{iq}$ = bobot nilai i pada data uji

$m$ = banyaknya jumlah nilai

c. Mengurutkan hasil penghitungan kedekatan ketetanggaan atau *similarity* ke dalam kelompok yang mempunyai kedekatan ketetanggaan atau *similarity*.

#### *2.2.5.4 Proses Neighbor Weighted K-Nearest Neighbor (NWKNN)*

Proses penghitungan dengan menggunakan metode KNN dan NWKNN hampir sama, namun pada metode NWKNN ada tahap tambahan lain. Perbedaan NWKNN dengan KNN ialah adanya penghitungan bobot kelas pada metode NWKNN yang membuat metode ini memiliki hasil yang lebih baik dibanding dengan KNN. Metode NWKNN muncul karena adanya ketidakseimbangan jumlah masing-masing kelas pada data latih. Pada penelitian yang dilakukan oleh Fadila (2016) diketahui bahwa akurasi yang didapatkan oleh KNN masih kurang tinggi. Setelah dilakukan penelitian dengan menggunakan metode NWKNN, metode NWKNN ini memiliki hasil akurasi yang lebih baik dibanding KNN dikarenakan adanya proses pembobotan pada tiap kelas sehingga membantu mengetahui jenis yang berasal dari kelas minoritas pada data. Penghitungan bobot dapat ditunjukkan pada Persamaan 2.2 (Tan, 2005).

$$\text{weight}_i = \frac{1}{\left(\frac{Num(c_i^d)}{\text{Min}\{Num(c_j^d)|n=1,\ldots,k*}\right)^{1/exp}} \quad (2.2)$$

Di mana

$Num(c_i^d)$ = banyaknya data latih $d$ pada kelas $i$

$Num(c_j^d)$ = banyaknya data latih $d$ pada kelas $j$, di mana $j$ terdapat dalam himpunan $k$ tetangga terdekat

$exp$ = eksponen (nilai $exp$ lebih dari 1)

Nilai $exp$ digunakan untuk mempengaruhi nilai bobot pada masing-masing kelas, yaitu untuk menambah nilai bobot itu sendiri. Setiap nilai bobot yang telah dihitung digunakan untuk menghitung nilai skor data uji terhadap setiap kelas. Penghitungan skor pada metode NWKNN hampir sama dengan KNN, namun pada penghitungan skor pada metode NWKNN setiap jumlah skor kelas dikalikan dengan bobot masing-masing kelas.

Penghitungan skor pada metode NWKNN dapat dilakukan dengan persamaan (Tan, 2005):

$$\text{skor}(X, C_i) = \text{weight}_i * \left(\sum_{dj\text{NWKNN}(X)} \left(\left(\sqrt{\sum_{i=1}^{n}(x_{2i} - x_{1i})^2}\right) * \delta(d_j, C_i)\right)\right) \quad (2.3)$$

atau

$$\text{skor}(X, C_i) = \text{weight}_i * \left(\sum_{dj\text{NWKNN}(X)} \left(Sim(q, d_j) * \delta(d_j, C_i)\right)\right) \quad (2.4)$$

Di mana

$\text{weight}_i$ = bobot kelas $i$

$dj\text{NWKNN}(X)$ = data latih $d_j$, pada kumpulan tetangga terdekat dari data uji $X$

$\sqrt{\sum_{i=1}^{n}(x_{2i} - x_{1i})^2}$ = jarak antara data uji dan data latih

$\delta(d_j, C_i)$ = akan bernilai 1 jika nilai jarak $\in C_i$dan bernilai 0 jika nilai jarak $\notin C_i$

$Sim(q, d_j)$ = nilai CosSim antara data uji dan data latih

$C_i$ = jenis atau kelas $i$

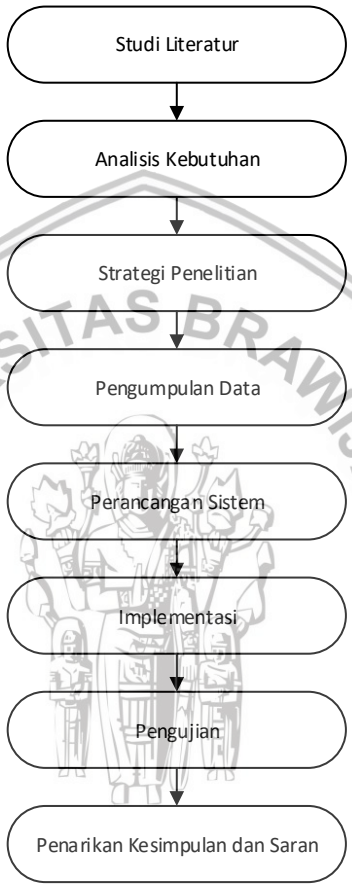
### *2.2.6 Akurasi* Sistem

Akurasi sistem ialah penghitungan untuk mengukur seberapa dekat suatu angka pengukuran terhadap angka sebenarnya. Penghitungan akurasi dilakukan dengan membagi jumlah data uji yang diklasifikasikan benar dengan jumlah total data uji dikalikan 100%, seperti persamaan berikut (Retwitasari, 2016):

$$\text{akurasi} = \frac{\text{jumlah data uji benar}}{\text{jumlah data}} \times 100\% \quad (2.5)$$

# BAB 3 METODOLOGI

Bab metodologi menjelaskan langkah-langkah yang dilakukan pada saat melakukan penelitian. Dalam bab ini akan dijelaskan tentang tahapan metode yang dilakukan dalam implementasi metode *Neighbor Weighted K-Nearest Neighbor (NWKNN)* pada klasifikasi jenis kanker berdasarkan struktur protein. Tahap-tahap penelitian dapat digambarkan pada diagram blok pada Gambar 3.1.




**Gambar 3.1 Diagram Blok Metodologi Penelitian**

## 3.1 Studi Literatur

Pada tahap ini dilakukan pembelajaran studi maupun pustaka pada bidang ilmu yang berhubungan dengan penelitian yang dijalankan, yaitu yang mengenai klasifikasi jenis kanker menggunakan metode *Neighbor Weighted K-Nearest Neighbor* (NWKNN), yaitu di antaranya:

a. Metode *Neighbor Weighted K-Nearest Neighbor* (NWKNN)
b. Penyakit kanker

Literatur yang digunakan didapat dari berbagai macam sumber seperti buku, artikel serta jurnal penelitian yang pernah dilaksanakan sebelumnya.

## 3.2 Analisis Kebutuhan

Analisis kebutuhan berfungsi untuk mengetahui kebutuhan yang diperlukan dalam membangun sistem untuk mengklasifikasi jenis kanker berdasarkan struktur protein. Kebutuhan yang diperlukan adalah kebutuhan berupa perangkat keras, perangkat lunak, kebutuhan perangkat lunak penunjang dan kebutuhan data. Kebutuhan ini digunakan untuk acuan untuk merancang, implementasi dan melakukan pengujian sistem. Kebutuhan yang diperlukan dalam penelitian ini antara lain:

1. Kebutuhan perangkat keras:
    - Laptop dengan *processor* Intel(R) Core(TM) i5-4200U CPU @ 1.60GHz 2.30GHz
    - Kapasitas Memor (RAM) sebesar 4.00 GB
2. Kebutuhan perangkat lunak:
    - Sistem Operasi Windows 8 / 10
    - Netbeans IDE 8.2
3. Kebutuhan Data:
    - Data struktur protein

## 3.3 Strategi Penelitian

Pada penelitian ini, strategi penelitian yang diambil ialah berupa eksperimen, di mana eksperimen yang dilakukan ialah dengan menerapkan metode *Neighbor Weighted K-Nearest Neighbor (NWKNN)* untuk klasifikasi jenis penyakit kanker berdasarkan susunan protein. Penerapan ini dilakukan dengan cara menerapkannya pada program komputer agar bisa diakukan pengujian dan diketahui tingkat akurasinya serta mengetahui apakah metode yang digunakan cocok untuk diimplementasikan pada permasalahan tersebut.

## 3.4 Pengumpulan Data

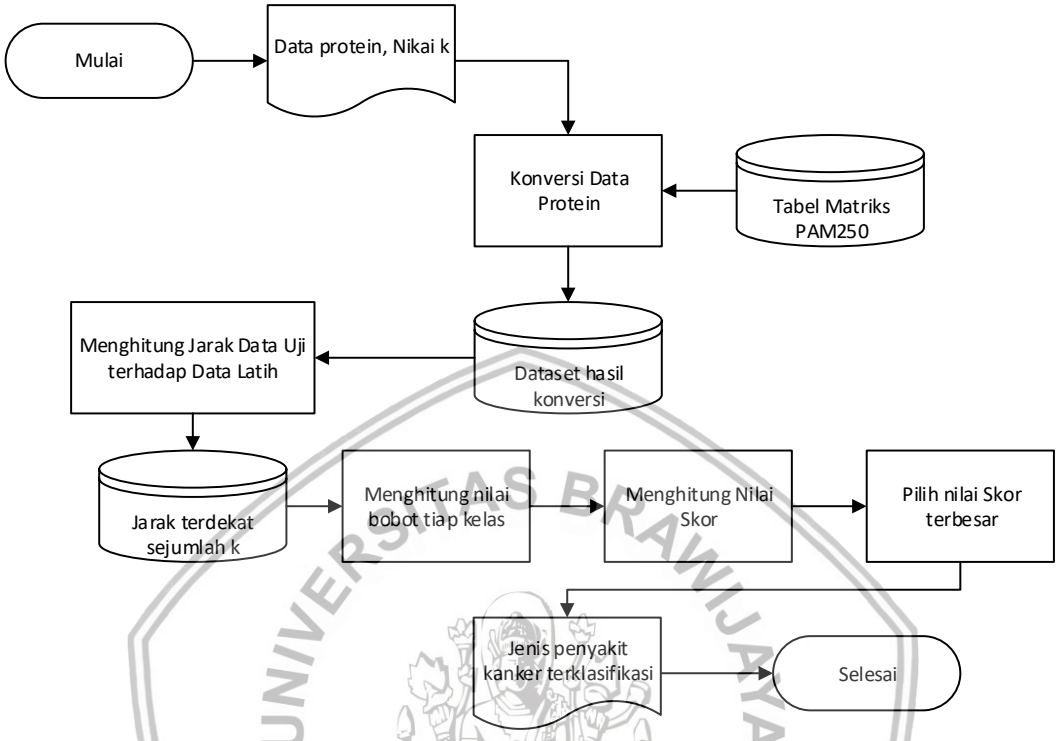
Pengumpulan data yang dilakukan merupakan pengumpulan data sekunder. Data sekunder diperoleh dari penelitian sebelumnya yang dilakukan oleh Tawang (2018). Data tersebut ialah data sekuens yang didapat dari http://www.uniprot.org dan database TP53 dari http://p53.free.fr/ .

Data kanker yang diklasifikasi ialah non-cancer, kanker payudara, kanker usus dan kanker paru-paru. Kelas-kelas yang digunakan ini berdasarkan ketersediaan serta keterbatasan data yang didapat dari sumber data serta keurgensian jenis kanker tersebut pada situasi yang terjadi di lingkungan sekitar penulis berdasarkan latar belakang yang telah dideskripsi.

## 3.5 Perancangan Sistem

Algoritma yang digunakan pada penelitian ini ialah *Neighbor Weighted K-Nearest Neighbor (NWKNN)*. Tahap awal algoritma ini ialah tahap *preprocessing*, yaitu tahap normalisasi data terhadap data latih serta data uji yang telah dimasukkan. Setelah itu dilakukan tahap algoritma *K-Nearest Neighbor (k-NN).*

Kemudian dapat digunakan proses proses penghitungan algoritma *Neighbor Weighted K-Nearest Neighbor (NWKNN)*. Alur perancangan algoritma *Neighbor Weighted K-Nearest Neighbor (NWKNN)* ditunjukkan pada Gambar 3.1.




**Gambar 3.2 Perancangan Algoritma NWKNN**

## 3.6 Implementasi

Pada tahap implementasi akan menjelaskan tentang implementasi metode *Neighbor Weighted K-Nearest Neighbor* (NWKNN) untuk klasifikasi jenis kanker berdasarkan perancangan manualisasi. Implementasi sistem menggunakan bahasa pemrograman Java beserta bantuan dengan menggunakan perangkat lunak penunjang yang telah dijelaskan di sub bab analisis kebutuhan.

## 3.7 Pengujian

Pada tahap ini dilakukan pengujian yang bertujuan untuk menunjukkan bahwa sistem dapat bekerja sesuai dengan yang diharapkan. Pengujian yang dapat dilakukan antara lain adalah:

1. Pengujian terhadap pengaruh perubahan jumlah data latih dan data uji
2. Pengujian terhadap pengaruh nilai K
3. Pengujian terhadap pengaruh nilai E

## 3.8 Penarikan Kesimpulan dan Saran

Kesimpulan dari penelitian ini dapat ditarik setelah penelitian selesai dilakukan. Kesimpulan yang diambil berdasarkan hasil implementasi dan pengujian yang telah dilakukan. Kesimpulan pada penelitian ini diharapkan dapat

dimanfaatkan dan dikembangkan pada bidang yang bersangkutan, serta saran digunakan untuk memberi pandangan terhadap penelitian selanjutnya.

# BAB 4 PERANCANGAN

Bab ini membahas tentang perancangan pada "Klasifikasi Jenis Kanker Berdasarkan Struktur Protein Menggunakan "*Neighbor Weighted K-Nearest Neighbor (NWKNN)*". Pohon perancangan pada sistem ini meliputi tiga tahap yaitu analisis kebutuhan yang terdiri atas deskripsi sistem, analisis kebutuhan data, dan identifikasi aktor. Tahap selanjutnya ialah tahap perancangan perangkat lunak yang terdiri dari perancangan algoritma dan perhitungan manual. Tahap yang terakhir adalah perancangan pengujian terdiri dari pengujian pengaruh jumlah cluster dan pengujian pengaruh jumlah dataset.

## 4.1 Analisis Kebutuhan

Tahap awal perancangan sistem kalsifikasi jenis kanker berdasarkan struktur protein menggunakan metode *Neighbor Weighted K-Nearest Neighbor (NWKNN)* adalah analisis kebutuhan perangkat lunak. Analisis ini terbagi menjadi dua, yaitu deksripsi sistem yang dibangun dan analisis kebutuhan data yang digunakan dalam sistem klasifikasi.

### 4.1.1 Deskripsi Sistem

Penelitian ini bertujuan untuk membuat aplikasi berbasis desktop yang berfungsi melakukan klasifikasi jenis kanker berdasarkan data yang digunakan. Data yang digunakan ialah struktur protein pada tubuh. Data yang digunakan dibagi menjadi tiga, yaitu data latih, data uji dan data *wild*. Data *wild* merupakan data acuan yang digunakan sebagai proses konversi dataset protein dari data yang bertipe string menjadi data bertip integer agar dapat dilakukan proses penghitungan klasifikasi menggunakan *Neighbor Weighted K-Nearest Neighbor (NWKNN)*. Data uji merupakan dataset protein yang akan dilakukan proses klasifikasi.

### 4.1.2 Analisis Kebutuhan Data

Penelitian ini menggunakan tiga jenis dataset, yaitu data latih, data uji dan data *wild*. Data *wild* digunakan sebagai data acuan yang digunakan sebagai bahan perbandingan dalam proses konversi data total menggunakan matriks PAM250, data latih merupakan dataset protein yang telah diketahui identifikasi kelasnya dan merupakan data yang akan diolah, dan data uji merupakan data yang digunakan untuk menguji akurasi sistem dengan menggunakan metode *Neighbor Weighted K-Nearest Neighbor (NWKNN).*

**Tabel 4.1 Sampel Data Protein**

| Data Protein | Kelas |
|---|---|
| MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDD IEQWFTEDPGPDEAPRMPEAAPPVAPAPAAPTPAAPAPAPSWPLSSSV PSQKTYQGSYGFRLGFLHSGTAKSVTCTYSPALNKMFCQLAKTCPVQLW VDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLAPPQHL | *Non-Cancer* |

| | |
|---|---|
| IRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCM<br>GGMNRRPILTIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGE<br>PHHELPPGSTKRALPNNTSSSPQPKKKPLDGEYFTLQIRGRERFEMFRELN<br>EALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSRHKKLMFKTEGPDS | |
| MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDD<br>IEQWFTEDPGPDEAPRMPEAAPPVAPAPAAPTPAAPAPAPSWPLSSSV<br>PSQKTYQGSYGFRLGFLHSGTAKSVTCTYSPALNKMFCQLAKTCPVQLW<br>VDSTPPPGTRVRAMAIYKQSQHMTEVVRRCRHHERCSDSDGLAPPQHL<br>IRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCM<br>GGMNRRPILTIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGE<br>PHHELPPGSTKRALPNNTSSSPQPKKKPLDGEYFTLQIRGRERFEMFRELN<br>EALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSRHKKLMFKTEGPDSD | *Breast Cancer* |
| MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDD<br>IEQWFTEDPGPDEAPRMPEAAPPVAPAPAAPTPAAPAPAPSWPLSSSV<br>PSQKTYQGSYGFRLGFLHSGTAKSVTCTYSPALNKMFCQLAKTCPVQLW<br>VDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHEPCSDSDGLAPPQHLI<br>RVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMG<br>GMNRRPILTIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEP<br>HHELPPGSTKRALPNNTSSSPQPKKKPLDGEYFTLQIRGRERFEMFRELNE<br>ALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSRHKKLMFKTEGPDSD | *Colorectal Cancer* |
| MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDD<br>IEQWFTEDPGPDEAPRMPEAAPPVAPAPAAPTPAAPAPAPSWPLSSSV<br>PSQKTYQGSYGFRLGFLHSGTAKSVTCTYSPALNKMFCQLAKTCPVQLW<br>VDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPNHERCSDSDGLAPPQHL<br>IRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCM<br>GGMNRRPILTIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGE<br>PHHELPPGSTKRALPNNTSSSPQPKKKPLDGEYFTLQIRGRERFEMFRELN<br>EALELKDAQAGKEPGGSRAHSSHLKSKKGQSTSRHKKLMFKTEGPDSD | *Lung Cancer* |

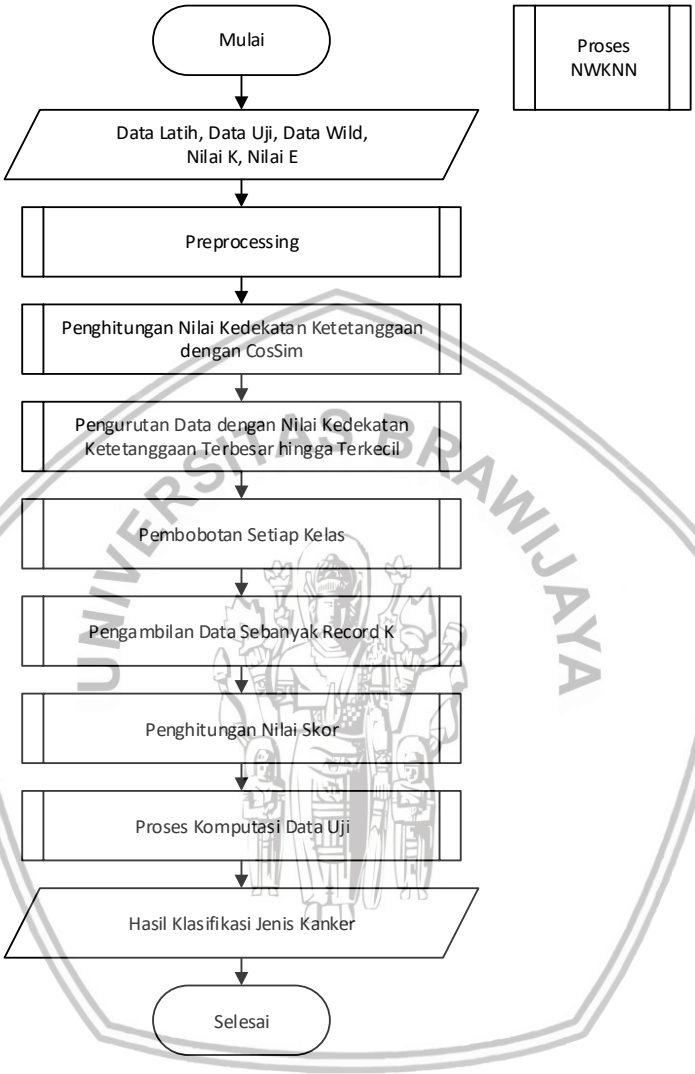
Terdapat empat kelas dalam dataset protein, yaitu kelas *Non-Cancer* (NC) atau tidak kanker yang disimbolkan dengan 0, kelas *Breast Cancer* (BC) atau kanker payudara yang disimbolkan dengan 1, kelas *Colorectal Cancer* (CC) atau kanker usus yang disimbolkan dengan 2, dan kelas *Lung Cancer* (LC) atau kanker paru-paru yang disimbolkan dengan 3.

## 4.2 Perancangan Perangkat Lunak

Proses perancangan perangkat lunak diperlukan ketika sedang membangun sebuah sistem. Hal ini memiliki tujuan untuk dapat dijadikan sebagai acuan dan/atau panduan dalam proses pembangunan sistem. Perancangan perangkat lunak yang harus dilakukan adalah perancangan algoritma, perhitungan manual, dan perancangan antarmuka.

### 4.2.1 Perancangan Algoritma

Proses pengklasifikasian jenis kanker berdasarkan struktur protein dapat dilakukan setelah data yang dibutuhkan pada sistem tersedia. Perancangan algoritma ditampilkan dalam Gambar 4.1.



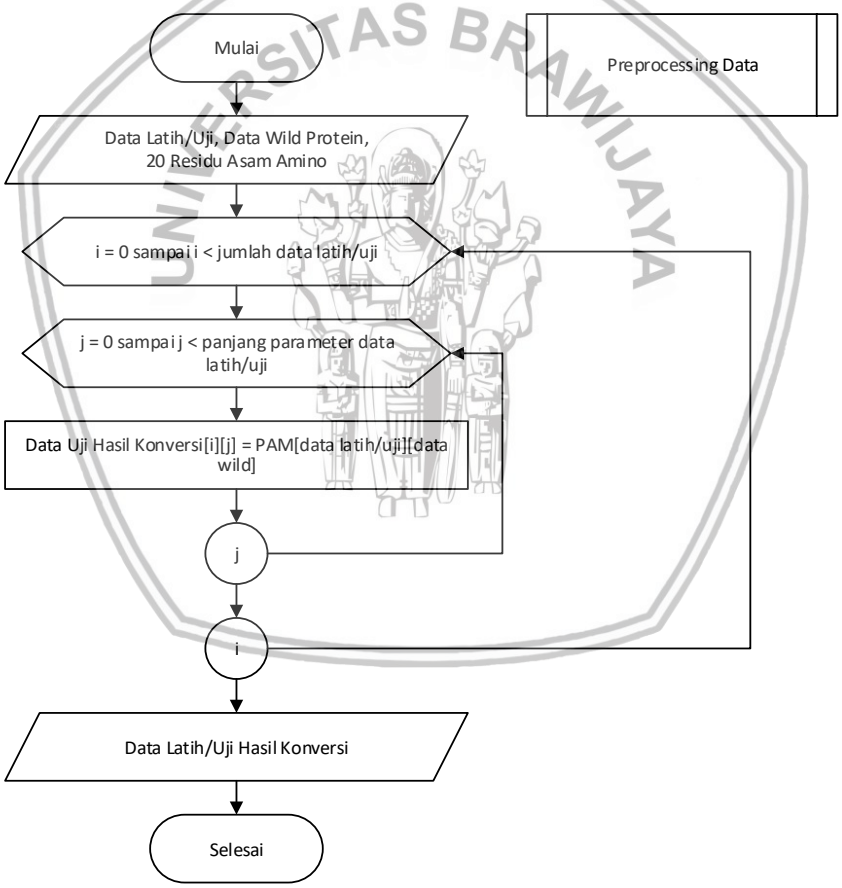

**Gambar 4.1 Diagram Alir Sistem**

Proses input data merupakan proses berupa memasukkan dataset protein (data latih) yang telah disiapkan, data uji protein yang disiapkan dan data *wild*, dengan syarat panjang string antara data uji, data latih, data *wild* adalah sama panjangnya. Selain dataset tersebut, sistem juga memerlukan data K dan E di mana K digunakan untuk pengambilan data record dan E merupakan *Exponent* yang akan mempengaruhi nilai bobot kelas. Proses pertama yang dilakukan sistem ialah *preprocessing*, yaitu mengkonversi/mengubah data fisik berupa karakter menjadi data numerik bertipe data integer menggunakan tabel PAM250 agar dapat dilakukan perhitungan. Lalu dilakukan penghitungan nilai kedekatan ketetanggaan menggunakan *Cosine Similiarity* seperti pada persamaan 2.1. Setelah didapatkan masing-masing nilai kedekatan ketetanggaan pada setiap data, nilai kedekatan ketetanggaan tersebut diurutkan dari yang terbesar hingga terkecil. Lalu dilakukan

penghitungan bobot masing-masing kelas dengan menggunakan persamaan 2.2. Setelah proses penghitungan bobot berhasil diakukan, sistem akan mengambil data yang telah terurut sebanyak nilai K yang telah dimasukkan sebelumnya. Kemudian dilakukan penghitungan nilai skor dengan menggunakan persamaan 2.3. Lalu dilakukan proses komputasi data uji, di mana nilai terbesar dari skor kelas merupakan hasil kelas klasifikasi data uji.

### 4.2.2 Proses *Preprocessing*

Proses *preprocessing* bertujuan untuk mengubah/mengkonversi data protein yang semula bertipe data string menjadi integer. Hal ini dilakukan agar data bisa diolah dan dilakukan penghitungan kedekatan ketetanggaan dan dapat dilakukan penghitungan dengan metode yang digunakan, yaitu metode *Neighbor Weighted K-Nearest Neighbor (NWKNN).* Proses konversi ini dilakukan berdasarkan tabel PAM250. Proses *preprocessing* dapat digambarkan melalui diagram alir pada Gambar 4.2.



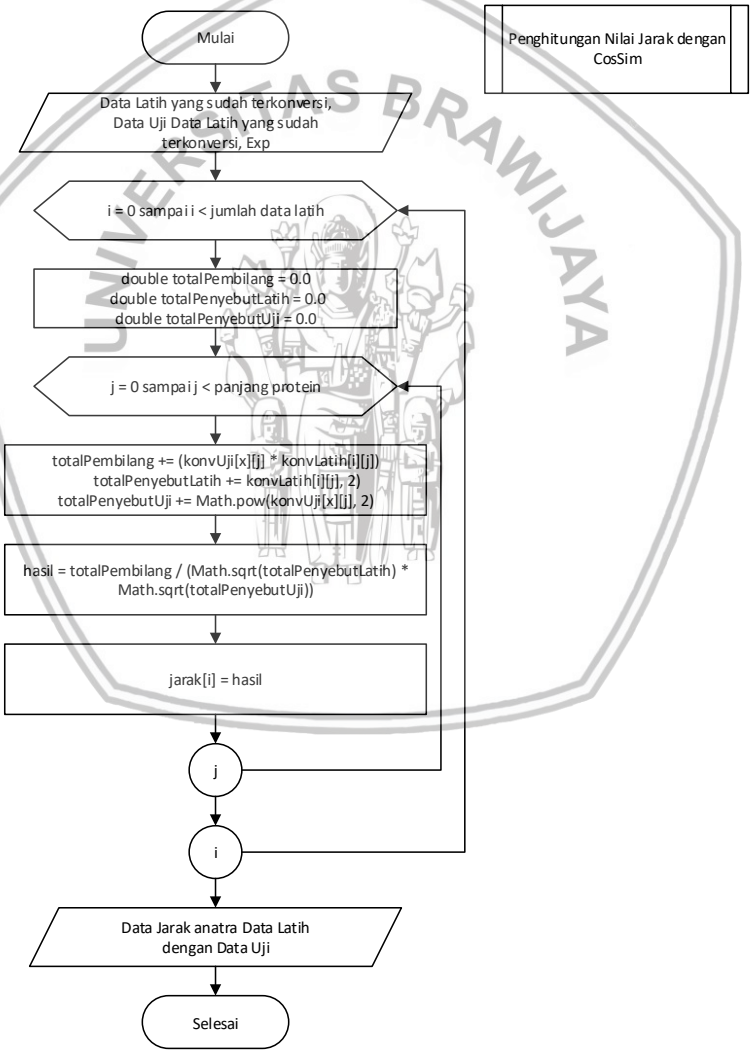

**Gambar 4.2 Diagram Alir Proses *Preprocessing* Data**

Langkah-langkah proses preprocessing baik data latih dan data uji ialah dengan memasukkan data uji, data latih, data *wild* dan 20 residu asam amino. Data *wild* digunakan untuk pembandingan masing-masing parameter data uji dan data latih untuk dicocokkan dengan 20 residu asam amino agar didapatkan nilai numeriknya pada setiap string/parameter dari masing-masing data sehingga dapat dilakukan penghitungan. Kemudian dilakukan perulangan sebanyak jumlah data latih/data

uji dan perulangan sebanyak jumlah string/parameter data latih/uji. Di dalam perulangan inilah akan dicocokkan antara data *wild* dengan data uji/*wild* dan dicocokkan kembali dengan 20 residu asam amino berdasarkan pada Tabel 2.1. Dengan ini didapatkan data konversi berupa numerik dari masing-masing parameter/panjang karakter dari masing-masing data uji dan latih.

### 4.2.3 Proses Menghitung Nilai Kedekatan ketetanggaan dengan *Cosine Similiarity*

Setelah masing-masing data terkonversi menjadi tipe data integer, maka selanjutnya adalah menghitung nilai kedekatan ketetanggaan antara data latih dengan data uji. Penghitungan kedekatan ketetanggaan menggunakan penghitungan *Cosine Similiarity*. Algoritma penghitungan kedekatan ketetanggaan dapat ditunjukkan pada Gambar 4.3.

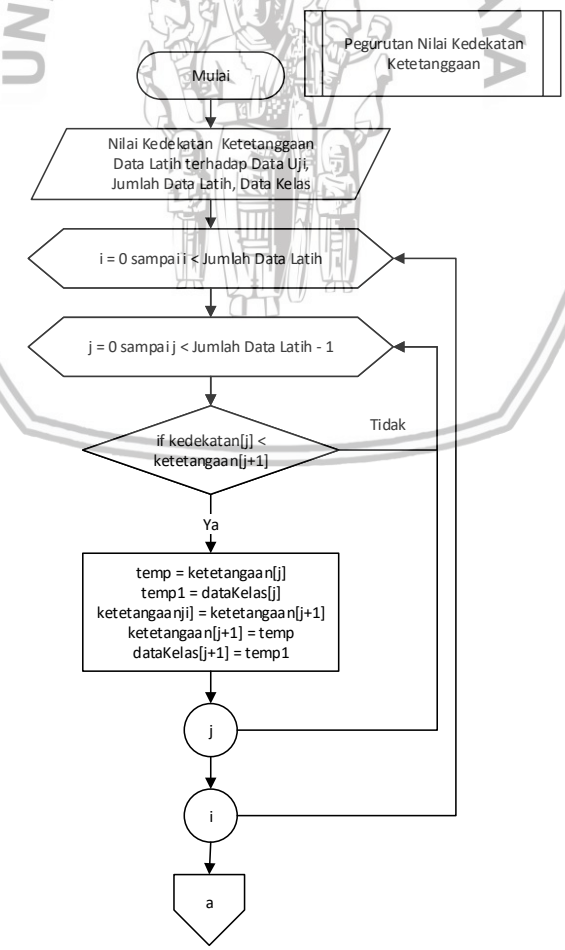


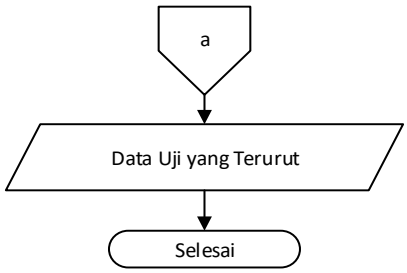

**Gambar 4.3 Diagram Alir Proses Penghitungan Nilai Kedekatan ketetanggaan Dengan *Cosine Similarity***

Langkah-langkah dari proses penghitungan kedekatan ketetanggaan dengan *Cosine Similiarity* ialah setelah data latih dan data uji terkonversi pada saat preprocessing, masing-masing data latih akan dihitung nilai kedekatan ketetanggaannya dengan data uji menggunakan *Cosine Similiarity* pada persamaan 2.1. Lalu dilakukan perulangan sebanyak data latih dan sebanyak panjang protein/parameter. Pada penghitungan ini dilakukan penghitungan terpisah. totalPembilang ialah penghitungan pembilang adalah variabel untuk menghitung pembilang dari *Cosine Similiarity*¸yaitu mengalikan masing-masing parameter, totalPenyebutLatih adalah variabel menghitung penyebut data latih yang berupa total akar dari kuadrat masing-masing parameter data latih, dan totalPenyebutUji ialah variabel menghitung penyebut data latih yang berupa total akar dari kuadrat masing-masing parameter data uji. Kemudian digunakan variabel hasil yang digunakan untuk melakukan penghitungan *Cosine Similiarity* secara keseluruhan.

### 4.2.4 Proses Mengurutkan Data dengan Kedekatan ketetanggaan Terbesar hingga Terkecil

Setelah didapatkan *Cosine Similiarity* dari masing-masing data latih, hasilnya diurutkan dari yang terbesar hingga yang terkecil. Algoritma pengurutan ini dapat ditunjukkan pada Gambar 4.4.



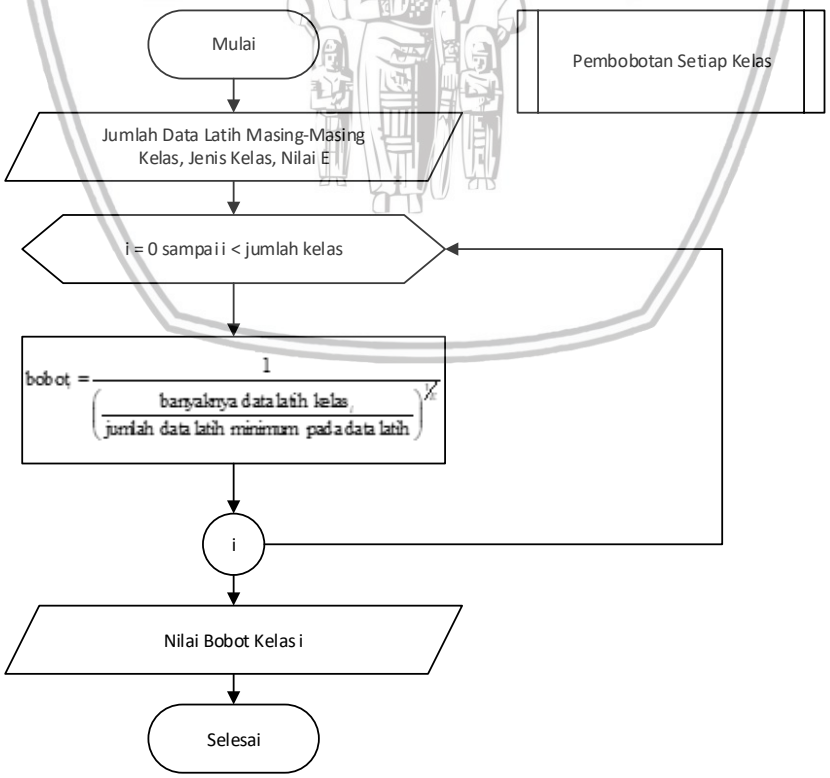



**Gambar 4.4 Diagram Alir Tahapan Proses Pengurutan Kedekatan ketetanggaan**

Langkah pengurutan kedekatan ketetanggaan ini dilakukan berupa pengurutan dari data dengan kedekatan ketetanggaan terbesar hingga terkecil. Proses ini dilakukan perulangan sebanyak jumlah data latih. Apabila nilai kedekatan ketetanggaan tertentu lebih kecil dibanding nilai kedekatan ketetanggaan pada urutan setelahnya, maka nilai kedekatan ketetanggaan tertentu tersebut digantikan dengan nilai kedekatan ketetanggaan setelahnya, dan nilai setelahnya digantikan dengan nilai tertentu tersebut. Penggantian ini dilakukan untuk saling memindahkan urutan data berdasarkan nilai kedekatan ketetanggaan yang terbesar pada urutan awal.

### 4.2.5 Proses Pembobotan Setiap Kelas

Dengan penghitungan bobot akan didapatkan hasil bobot dari setiap kelas. Proses penghitungan bobot akan dilakukan seperti persamaan. Proses penghitungan bobot dapat dilihat pada Gambar 4.5 berikut.

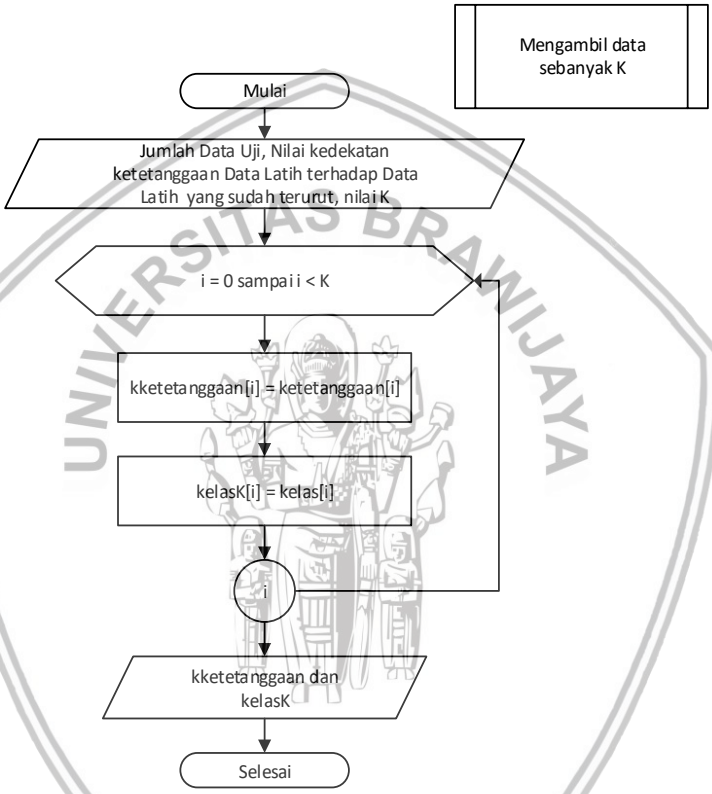



**Gambar 4.5 Diagram Alir Pembobotan Setiap Kelas**

Penghitungan bobot setiap kelas ini dilakukan dengan adanya jumlah data latih tiap kelas, jenis kelas serta nilai E (*Exponent*). Kemudian dilakukan perulangan sebanyak jumlah kelas yang ada. Lalu dilakukan penghitungan bobot kelas dengan melakukan penghitungan banyaknya data latih pada kelas tertentu dan dibagi dengan jumlah data minimum pada data latih dan dipangkat dengan satu per E.

### 4.2.6 Proses Pengambilan Data Sebanyak K

Berikutnya adalah proses mengambil data sebanyak nilai K yang telah diinput oleh pengguna. Pada penghitungan ini, K yang diambil adalah 5. Algoritma pengambilan data sebanyak nilai K ini dapat ditunjukkan pada Gambar 4.6.



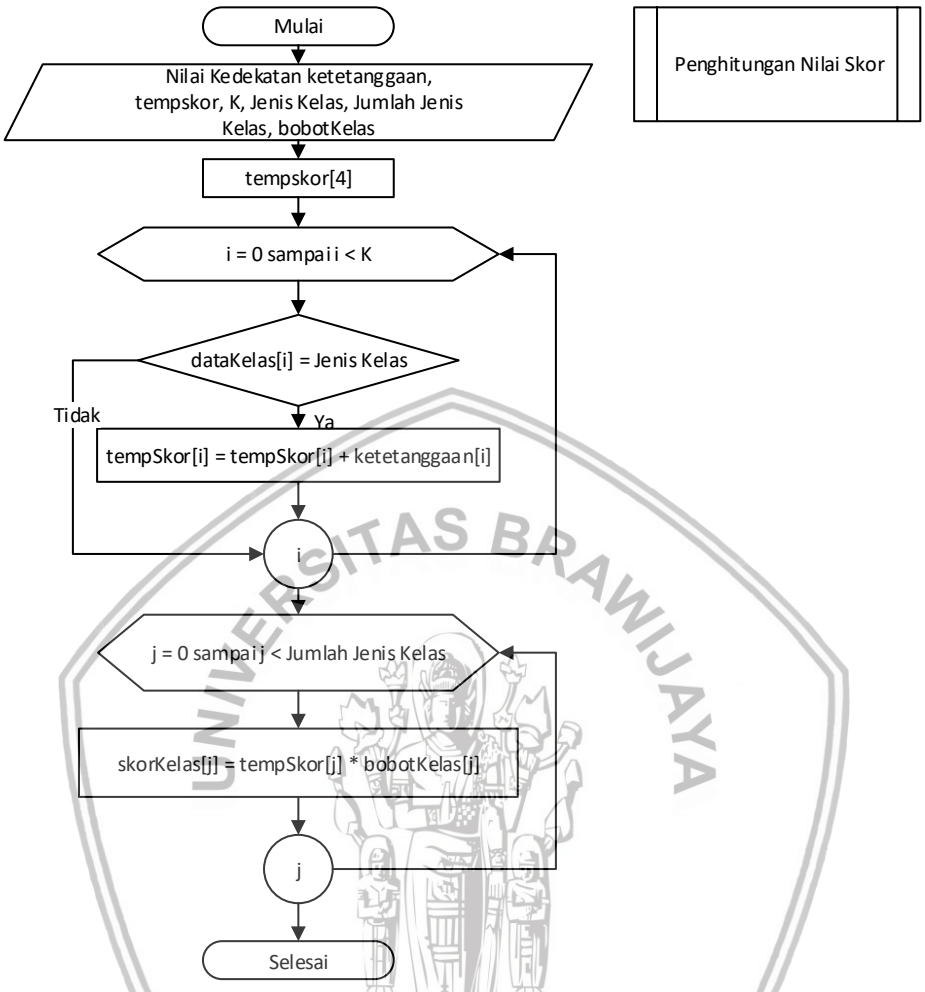

**Gambar 4.6 Diagram Alir Tahapan Proses Pengambilan Sebanyak K**

Pada proses ini, dilakukan perulangan sebanyak nilai K karena data yang akan diambil hanya sebanyak K untuk penghitungan skor. Kemudian menggunakan variabel kJarak dan kelasK untuk menyimpan kedekatan ketetanggaan dan kelas yang akan tersimpan untuk menghitung nilai skor.

### 4.2.7 Proses Penghitungan Nilai Skor

Tahap penghitungan nilai skor ini yang membedakan antara *K-Nearest Neighbor (KNN)* biasa dengan *Neighbor Weighted K-Nearest Neighbor (NWKNN)* karena pada NWKNN menggunakan cara penghitungan skor yang lebih kompleks dibandingkan dengan KNN biasa karena terdapat penghitungan bobot pada masing-masing kelas. Proses penghitungan nilai skor dilakukan pada setiap jenis yang ada sesuai dengan data hasil *Cosime Simmiliarity* yang telah diambil sebanyak

nilai K yang telah ditentukan sebelumnya. Algoritma penghitungan nilai skor ditunjukkan pada Gambar 4.7.

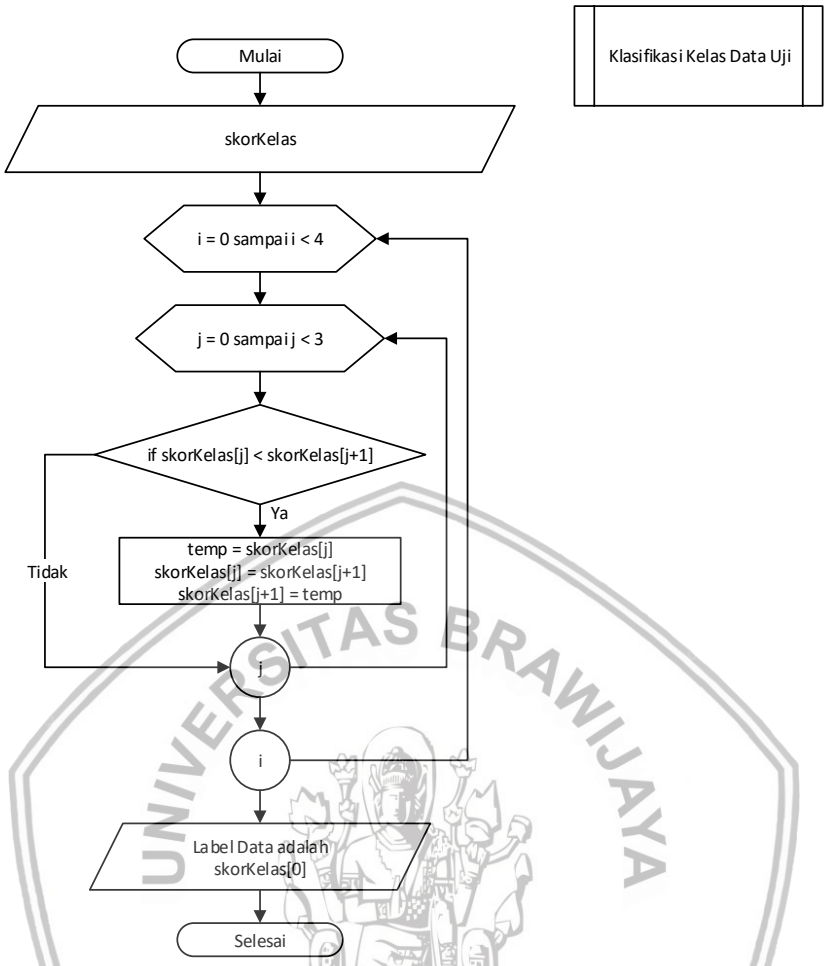



**Gambar 4.7 Diagram Alir Algoritma Penghitungan Nilai Skor**

Pada tahap ini menggunakan satu variabel bernama tempSkor yang digunakan sebagai variabel yang berisi jumlah kedekatan ketetanggaan pada masing-masing kelas. Langkah ini dilakukan sebanyak nilai K yang diinput. Setelah itu dilakukan perulangan kembali sebanyak jumlah jenis kelas untuk menghitung nilai masing-masing skor kelas dengan mengalikan antara tempSkor dengan bobot masing-masing kelas. Kemudian setelah keluar dari perulangan awal, dilakukan kembali perulangan baru sebanyak jumlah jenis kelas untuk menghitung masing-masing skor kelas dengan cara mengalikan masing-masing tempSkor dengan masing-masing bobot kelas.

### 4.2.8 Proses Komputasi Kelas Data Uji

Setelah dilakukan proses voting, kemudian proses yang dilakukan ialah labelisasi data uji untuk mengetahui kelas manakah yang akan menjadi kelas data uji. Proses ini juga digunakan sebagai hasil akhir. Algoritma komputasi labelisasi kelas data uji ditunjukkan pada Gambar 4.9.




**Gambar 4.8 Diagram Alir Algoritma Klasifikasi Kelas Data Uji**

Langkah-langkah komputasi kelas data uji adalah mengikutsertakan skor dari masing-masing kelas. Setelah itu dilakukan perulangan untuk dilakukan sorting. Di dalam penghitungan perulangan tersebut, dilakukan pengecekan kondisi apakah skor kelas tertentu lebih kecil dibanding skor kelas selanjutnya. Apabila skor kelas tertentu tersebut lebih kecil nilainya, maka skor kelas tertentu ini dipindahkan nilainya dengan skor kelas yang selanjutnya, dan skor kelas selanjutnya menjadi skor kelas tertentu. Hal ini akan membuat skor kelas tertentu akan terinisialisasi dengan skor kelas terbesar, karena kelas terklasifikasi yang diambil ialah skor kelas dengan nilai terbesar.

## 4.3 Penghitungan Manual

Pada sub-bab penghitungan manual ini diambil beberapa data (total 24 data terdiri dari 23 data latih dan 1 data uji) dari dataset protein yang digunakan. Satu data *wild* digunakan sebagai data perbandingan dalam proses konversi data dan 23 dataset protein yang diambil secara acak untuk data latih dari data total protein yang akan dilakukan proses klasifikasi mengunakan metode *Neighbor Weighted K-Nearest Neighbor (NWKNN)*. Pada penghitungan manual ini, data yang diambil hanya diambil 10 karakter panjang data saja dari 363 karakter dalam satu data.

Dari dataset tersebut telah diketahui kelas masing-masing yaitu *Non-Cancer* (NC), *Breast Cancer* (BC), *Colorecctal Cancer* (CC), dan *Lung Cancer* (LC).

### 4.3.1 Konversi Data

Dataset yang tersedia memiliki tipe data berupa string. Dataset harus dikonversikan ke tipe data integer terlebih dahulu sebelum dilakukan proses penghitungan. Pengkonversian data dilakukan dengan cara mencocokkan dataset dengan data *wild* yang diambil dari satu data teratas dari dataset. Pengkonversian data dilakukan dengan mengacu pada matriks PAM250. Berikut merupakan tabel data *wild* yang ditunjukkan pada Tabel 4.2, data latih yang ditunjukkan pada Tabel 4.3 dan data uji yang ditunjukkan pada Tabel 4.4.

**Tabel 4.2 Data *Wild bentuk Fisik***

| Data Protein | Variabel/Fitur | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 |
| Data Wild | R | C | P | H | H | E | R | C | S | D |

**Tabel 4.3 Data Latih *bentuk Fisik***

| Data Protein | Variabel/Fitur | | | | | | | | | | Kelas |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | |
| Data 1 | R | C | P | H | H | E | R | C | S | D | NC |
| Data 2 | G | C | P | H | H | E | R | C | S | D | NC |
| Data 3 | R | C | P | H | H | E | P | C | S | D | CC |
| Data 4 | R | C | P | H | R | E | R | C | S | D | BC |
| Data 5 | R | C | P | H | H | E | R | S | S | D | NC |
| Data 6 | R | C | P | H | H | E | R | C | S | D | LC |
| Data 7 | R | C | P | H | H | E | R | C | S | D | CC |
| Data 8 | R | C | P | N | H | E | R | C | S | D | LC |
| Data 9 | R | C | P | H | H | E | R | C | L | D | BC |
| Data 10 | R | S | P | H | H | E | R | C | S | D | NC |
| Data 11 | P | C | P | H | H | E | R | C | S | D | NC |
| Data 12 | R | C | P | H | H | E | R | Y | S | D | CC |
| Data 13 | R | C | P | H | H | E | R | A | S | D | NC |
| Data 14 | R | C | P | H | H | K | R | C | S | D | LC |
| Data 15 | R | F | P | H | H | E | R | C | S | D | CC |
| Data 16 | R | C | R | H | H | E | R | C | S | D | BC |
| Data 17 | C | C | P | H | H | E | R | C | S | D | LC |
| Data 18 | Q | C | P | H | H | E | R | C | S | D | NC |
| Data 19 | R | C | P | H | H | E | R | C | S | D | BC |
| Data 20 | R | C | P | H | H | E | R | C | S | N | CC |
| Data 21 | R | C | P | H | H | E | R | C | S | D | BC |
| Data 22 | R | C | L | H | H | E | R | C | S | D | LC |
| Data 23 | R | C | P | H | H | E | R | C | P | D | NC |

**Tabel 4.4 Data Uji *bentuk Fisik***

| Data Protein | Variabel/Fitur | | | | | | | | | | Kelas |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | |
| Data 1 | R | C | P | H | H | E | C | C | S | D | CC |

Hasil konversi data fisik menjadi data numerik untuk data latih ditampikan pada Tabel 4.5 dan ditampilkan pada Tabel 4.6 untuk data uji. Untuk kelas dikonversikan menjadi 0 bagi NC (*non-cancer)*, 1 bagi BC (*Breast Cancer*), 2 bagi CC (*Corolectal Cancer*) dan 3 bagi LC (*Lung Cancer*).

**Tabel 4.5 Data Latih Hasil Konversi**

| Data Protein | Variabel/Fitur | | | | | | | | | | Kelas |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | |
| Data 1 | 6 | 12 | 6 | 6 | 6 | 4 | 6 | 12 | 2 | 4 | 0 |
| Data 2 | -3 | 12 | 6 | 6 | 6 | 4 | 6 | 12 | 2 | 4 | 0 |
| Data 3 | 6 | 12 | 6 | 6 | 6 | 4 | 0 | 12 | 2 | 4 | 2 |
| Data 4 | 6 | 12 | 6 | 6 | 1 | 4 | 6 | 12 | 2 | 4 | 1 |
| Data 5 | 6 | 12 | 6 | 6 | 6 | 4 | 6 | 1 | 2 | 4 | 0 |
| Data 6 | 6 | 12 | 6 | 6 | 6 | 4 | 6 | 12 | 2 | 4 | 3 |
| Data 7 | 6 | 12 | 6 | 6 | 6 | 4 | 6 | 12 | 2 | 4 | 2 |
| Data 8 | 6 | 12 | 6 | 1 | 6 | 4 | 6 | 12 | 2 | 4 | 3 |
| Data 9 | 6 | 12 | 6 | 6 | 6 | 4 | 6 | 12 | -2 | 4 | 1 |
| Data 10 | 6 | 1 | 6 | 6 | 6 | 4 | 6 | 12 | 2 | 4 | 0 |
| Data 11 | 0 | 12 | 6 | 6 | 6 | 4 | 6 | 12 | 2 | 4 | 0 |
| Data 12 | 6 | 12 | 6 | 6 | 6 | 4 | 6 | 1 | 2 | 4 | 2 |
| Data 13 | 6 | 12 | 6 | 6 | 6 | 4 | 6 | -2 | 2 | 4 | 0 |
| Data 14 | 6 | 12 | 6 | 6 | 6 | 0 | 6 | 12 | 2 | 4 | 3 |
| Data 15 | 6 | -4 | 6 | 6 | 6 | 4 | 6 | 12 | 2 | 4 | 2 |
| Data 16 | 6 | 12 | 0 | 6 | 6 | 4 | 6 | 12 | 2 | 4 | 1 |
| Data 17 | -3 | 12 | 6 | 6 | 6 | 4 | 6 | 12 | 2 | 4 | 3 |
| Data 18 | 1 | 12 | 6 | 6 | 6 | 4 | 6 | 12 | 2 | 4 | 0 |
| Data 19 | 6 | 12 | 6 | 6 | 6 | 4 | 6 | 12 | 2 | 4 | 1 |
| Data 20 | 6 | 12 | 6 | 6 | 6 | 4 | 6 | 12 | 2 | 2 | 2 |
| Data 21 | 6 | 12 | 6 | 6 | 6 | 4 | 6 | 12 | 2 | 4 | 1 |
| Data 22 | 6 | 12 | -2 | 6 | 6 | 4 | 6 | 12 | 2 | 4 | 3 |
| Data 23 | 6 | 12 | 6 | 6 | 6 | 4 | 6 | 12 | 2 | 4 | 0 |

**Tabel 4.6 Data Uji Hasil Konversi**

| Data Protein | Variabel/Fitur | | | | | | | | | | Kelas |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | |
| Data 1 | 6 | 12 | 6 | 6 | 6 | 4 | -3 | 2 | 2 | 4 | 2 |

### 4.3.2 Penghitungan Nilai *Kedekatan Ketetanggaan*

Penghitungan kedekatan ketetanggaan menggunakan rumus *CosSimilarity* yang ditunjukkan pada Persamaan 2.1. Pada penghitungan ini menggunakan data 1 pada Tabel 4.5 sebagai data latih dan data pada Tabel 4.6 sebagai data uji.

$$\text{CosSim}(q, d_j) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|} = \frac{\sum_{i=1}^{m}(w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{m} {w_{ij}}^2 \cdot \sum_{i=1}^{m} {w_{iq}}^2}}$$

Pada kasus ini, $w_{ij}$ merupakan nilai bobot dari setiap asam amino dari data latih dari i=1 hingga i=10, dan $w_{iq}$ ialah nilai bobot setiap asam amino dari data uji dari i=1 hingga i=10.

$$\vec{d}_j \cdot \vec{q} = ((6 * 6) + (12 * 12) + (6 * 6) + (6 * 6) + (6 * 6) + (4 * 4) + \big(6 * (-3)\big) + (2 * 2) + (12 * 2) + (4 * 4))$$

$$\vec{d}_j \cdot \vec{q} = 450$$

$$|\vec{d}_j| \cdot |\vec{q}| = \sqrt{\big((6^2) + (12^2) + (6^2) + (6^2) + (6^2) + (4^2) + (6^2) + (2^2) + (12^2) + (4^2)\big)} \times \sqrt{(6^2) + (12^2) + (6^2) + (6^2) + (6^2) + (4^2) + ((-3)^2) + (2^2) + (2^2) + (4^2))}$$

$$|\vec{d}_j| \cdot |\vec{q}| = 490.31$$

$$\text{CosSim}(q, d_j) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|} = \frac{\sum_{i=1}^{m}(w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{m} {w_{ij}}^2 \cdot \sum_{i=1}^{m} {w_{iq}}^2}} = \frac{310}{350{,}24} = 0.917778873$$

Perhitungan yang sama juga dilakukan pada setiap data latih yang digunakan terhadap data uji. Hasil perhitungan *CosSim* secara keseluruhan antara data latih terhadap data uji ditunjukkan pada Tabel 4.7.

**Tabel 4.7 Nilai Kedekatan ketetanggaan Antara Data Latih dengan Data Uji**

| Data | Nilai CosSim | Kelas |
|---|---|---|
| Data 1 | 0.917778873 | 0 |
| Data 2 | 0.830188679 | 0 |
| Data 3 | 0.990521113 | 2 |
| Data 4 | 0.887981013 | 1 |
| Data 5 | 0.766327357 | 0 |
| Data 6 | 0.917778873 | 3 |
| Data 7 | 0.917778873 | 2 |
| Data 8 | 0.887981013 | 3 |
| Data 9 | 0.901462804 | 1 |
| Data 10 | 0.766327357 | 0 |

| | | |
|---|---|---|
| Data 11 | 0.876230215 | 0 |
| Data 12 | 0.766327357 | 2 |
| Data 13 | 0.676767083 | 0 |
| Data 14 | 0.899540309 | 3 |
| Data 15 | 0.609209397 | 2 |
| Data 16 | 0.876230215 | 1 |
| Data 17 | 0.830188679 | 3 |
| Data 18 | 0.887981013 | 0 |
| Data 19 | 0.917778873 | 1 |
| Data 20 | 0.912390025 | 2 |
| Data 21 | 0.917778873 | 1 |
| Data 22 | 0.847219346 | 3 |
| Data 23 | 0.917778873 | 0 |

### 4.3.3 Mengurutkan Nilai Kedekatan Ketetanggaan

Setelah didapatkan hasil kedekatan ketetanggaan masing-masing data latih terhadap data uji, masing-masing data diurutkan dari nilai terbesar hingga ke nilai yang terkecil berdasarkan nilai *CosSim*. Dataset yang telah diurutkan ditunjukkan pada Tabel 4.8.

**Tabel 4.8 Dataset Nilai Ketetanggaan yang telah diurutkan**

| Data | Nilai CosSim | Kelas |
|---|---|---|
| Data 3 | 0.990521113 | 2 |
| Data 1 | 0.917778873 | 0 |
| Data 6 | 0.917778873 | 3 |
| Data 7 | 0.917778873 | 2 |
| Data 19 | 0.917778873 | 1 |
| Data 21 | 0.917778873 | 1 |
| Data 23 | 0.917778873 | 0 |
| Data 20 | 0.912390025 | 2 |
| Data 9 | 0.901462804 | 1 |
| Data 14 | 0.899540309 | 3 |
| Data 4 | 0.887981013 | 1 |
| Data 8 | 0.887981013 | 3 |
| Data 18 | 0.887981013 | 0 |
| Data 11 | 0.876230215 | 0 |
| Data 16 | 0.876230215 | 1 |
| Data 22 | 0.847219346 | 3 |
| Data 2 | 0.830188679 | 0 |
| Data 17 | 0.830188679 | 3 |
| Data 5 | 0.766327357 | 0 |
| Data 10 | 0.766327357 | 0 |

| | | |
|---|---|---|
| Data 12 | 0.766327357 | 2 |
| Data 13 | 0.676767083 | 0 |
| Data 15 | 0.609209397 | 2 |

### 4.3.4 Pembobotan Setiap Kelas

Proses berikutnya ialah penghitungan bobot untuk setiap jenis kanker yang akan diklasifikasi dengan menggunakan nilai K dan nilai eksponen yang telah ditentukan. Penghitungan pembobotan setiap jenis kelas kanker dilakukan dengan menggunakan Persamaan 2.2. Berikut contoh penghitungan bobot untuk setiap kelas kanker dengan menggunakan nilai K=5 dan E=4.

Jumlah data pada masing-masing kelas kanker yang ada pada data latih meliputi:

- *Non Cancer* berjumlah 8
- *Breast Cancer* berjumlah 5
- *Colorectal Cancer* berjumlah 5
- *Lung Cancer* berjumlah *5*

Sesuai dengan data latih yang ada, *Non Cancer* merupakan kelas mayoritas dan sisanya merupakan kelas minoritas, maka:

$$\text{weight}_i = \frac{1}{\left(\frac{Num(c_i^d)}{\text{Min}\{Num(c_j^d)|n = 1, \dots, k*\}}\right)^{1/exp}}$$

$$\text{weight}_{(\text{non cancer})} = \frac{1}{\left(\frac{8}{5}\right)^{1/4}} = 0{,}8891$$

$$\text{weight}_{(\text{breast cancer})} = \frac{1}{\left(\frac{5}{5}\right)^{1/4}} = 1$$

$$\text{weight}_{(\text{colorectal cancer})} = \frac{1}{\left(\frac{5}{5}\right)^{1/4}} = 1$$

$$\text{weight}_{(\text{lung cancer})} = \frac{1}{\left(\frac{5}{5}\right)^{1/4}} = 1$$

Nilai pembobotan tersebut berlaku untuk data uji terhadap data latih (1-23). Berikut hasil pembobotan tiap kelas kanker dengan jumlah 23 data dapat dilihat pada Tabel 4.9.

**Tabel 4.9 Nilai bobot setiap kelas kanker**

| No | Jenis | Nilai Bobot |
|---|---|---|
| 1 | Non Cancer | 0,8891 |
| 2 | Breast Cancer | 1 |
| 3 | Colorectal Cancer | 1 |
| 4 | Lung Cancer | 1 |

### 4.3.5 Penghitungan Skor

Setelah didapatkan nilai bobot masing-masing kelas kanker, dilakukan penghitungan skor pada setiap kelas kanker yang termasuk dalam K tetangga untuk mengetahui hasil identifikasi kelas kanker dari data uji. Nilai skor terbesar akan menjadi hasil identifikasi. Sebelum melakukan penghitungan, data terlebih dahulu diambil sebanyak nilai K yang telah diurutkan sebelumnya berdasarkan nilai *CosSim* nya. Berikut contoh penghitungan skor berdasarkan Persamaan 2.4 pada saat K=5 ditunjukkan pada Tabel 4.10.

**Tabel 4.10 Dataset Sebanyak Record K**

| Data Protein | Nilai CosSim | Kelas |
|---|---|---|
| Data 3 | 0,986556521 | CC |
| Data 1 | 0,885107563 | NC |
| Data 6 | 0,885107563 | LC |
| Data 7 | 0,885107563 | CC |
| Data 19 | 0,885107563 | BC |

Setelah didapatkan data sebanyak record K, penghitungan nilai skor dapat dilakukan. Berdasarkan tabel nilai *CosSim* yang telah diurutkan dan diambil sebanyak nilai K=5, kelas yang diperoleh ialah *colorectal cancer, non cancer, lung cancer* dan *breast cancer*, maka skor yang dihitung pada data uji adalah skor keempat kelas tersebut.

Skor(non cancer) = 0,8891 * ((0,986556521 * 0) + (0,885107563 * 1) + ( 0,885107563 * 0) + (0,885107563 * 0) + (0,885107563 * 0) = **0.8160**

Skor(breast cancer) = 1 * ((0,986556521 * 0) + (0,885107563 * 0) + ( 0,885107563 * 0) + (0,885107563 * 0) + (0,885107563 * 1) = **0.9177**

Skor(colorectal cancer) = 1 * ((0,986556521 * 1) + (0,885107563 * 0) + ( 0,885107563 * 0) + (0,885107563 * 1) + (0,885107563 * 0) = **1.9082**

Skor(lung cancer) = 1 * ((0,986556521 * 0) + (0,885107563 * 0) + ( 0,885107563 * 1) + (0,885107563 * 0) + (0,885107563 * 0) = **0.9177**

Berdasarkan penghitungan nilai skor di atas dapat disimpulkan bahwa data uji tersebut teridentifikasi masuk ke dalam kelas *colorectal cancer* karena nilai terbesar dari penghitungan skor ketika K=5 adalah jenis *colorectal cancer*. Berdasarkan data asli dengan hasil penghitungan manual, keduanya menunjukkan hasil kelas yang sama, yaitu teridentifikasi *colorecctal cancer*.

## 4.4 Teknik Pengujian

Pengujian dilakukan untuk mengetahui tingkat akurasi terhadap nilai pada pengujian. Terdapat beberapa parameter pengujian dalam penelitian ini, yaitu pengujian terhadap pengaruh nilai K, pengujian terhadap pengaruh nilai E dan pengujian pengaruh perubahan jumlah data latih.

### 4.4.1 Pengujian Terhadap Pengaruh Perubahan Jumlah Data Latih dan Data Uji

Pengujian pengaruh perubahan jumlah data latih dan data uji dilakukan untuk mengetahui pengaruh penambahan dan pengurangan jumlah data latih dan data uji terhadap tingkat akurasi yang dihasilkan dengan metode yang digunakan. Pengujian dilakukan sebanyak tiga kali pada masing-masing persentase data latih dengan nilai K yang digunakan adalah 5, 10 dan 15. Nilai E yang digunakan adalah sebesar 3. Pengujian dilakukan dengan perbandingan jumlah data latih dan data uji sebanyak 90%:10%, 80%:20%, 70%:30%, 60%:40%, 50%:50%, 40%:60%, 50%:50, 40%:60%, 30%:70%, 20%:80% dan 10%:90% dari total dataset. Gambaran Tabel Pengujian pengaruh perubahan jumlah data latih ditunjukkan pada Tabel 4.11.

**Tabel 4.11 Tabel pengujian terhadap pengaruh perubahan jumlah data latih**

| Nilai E | Persentase Data Latih | Persentase Data Uji | Akurasi (%) | | | |
|---|---|---|---|---|---|---|
| | | | K=5 | K=10 | K=15 | Rata-Rata |
| 3 | 90 | 10 | | | | |
| | 80 | 20 | | | | |
| | 70 | 30 | | | | |
| | 60 | 40 | | | | |
| | 50 | 50 | | | | |
| | . | . | | | | |
| | . | . | | | | |
| | . | . | | | | |
| | 10 | 90 | | | | |

### 4.4.2 Pengujian Terhadap Pengaruh Nilai K

Pengujian dilakukan untuk mengetahui tingkat akurasi terhadap data latih dan data uji. Pengujian dilakukan dengan menggunakan jumlah dataset sebanyak 752 data. Perbandingan data latih dan data uji yang digunakan ialah perbandingan data latih dan data uji yang menghasilkan akurasi terbaik yang didapat dari Pengujian Terhadap Pengaruh Perubahan Jumlah Data Latih. Nilai K yang digunakan untuk pengujian sebesar 2 hingga 15 dengan nilai E yang tetap, yaitu sebesar 3. Gambaran tabel pengujian pengaruh nilai K ditunjukkan pada Tabel 4.12.

**Tabel 4.12 Tabel pengujian terhadap pengaruh nilai K**

| Persentase Data Latih | Persentase Data Uji | Nilai E | Nilai K | Akurasi Data Latih (%) |
|---|---|---|---|---|
| | | 3 | 2 | |
| | | | 3 | |

| | | | | |
|---|---|---|---|---|
| | | | 4 | |
| | | | . | |
| | | | . | |
| | | | 14 | |
| | | | 15 | |

### 4.4.3 Pengujian Terhadap Pengaruh Nilai E

Pengujian pengaruh nilai E dilakukan untuk mengetahui pengaruh nilai E terhadap akurasi metode apabila nilai E diubah. Pengujian dilakukan dengan mengubah nilai E secara acak, di mulai dari nilai E=2, E=3, E=4, E=5, E=6. Masing-masing nilai E akan dilakukan pengujian sebanyak 3 kali terhadap data uji dengan nilai K yang tetap dan data latih yang berbeda-beda. Perbandingan dataset yang digunakan ialah dataset terbaik yang didapat dari Pengujian Terhadap Pengaruh Perubahan Jumlah Data Latih. Nilai K yang digunakan adalah nilai K yang memberikan hasil akurasi maksimum pada pengujian yang dilakukan terhadap pengaruh nilai K. Rasio data latih dan data uji yang digunakan ialah rasio dengan nilai akurasi terbaik pada pengujian sebelumnya. Gambaran tabel pengujian pengaruh nilai E ditunjukkan pada Tabel 4.13.

**Tabel 4.13 Tabel pengujian terhadap pengaruh nilai E**

| Persentase Data Latih (%) | Persentase Data Uji (%) | Nilai K | Nilai E | Akurasi Data Latih (%) |
|---|---|---|---|---|
| | | 8 | 2 | |
| | | | 3 | |
| | | | 4 | |
| | | | 5 | |
| | | | 6 | |

### 4.4.4 Pengujian Perbandingan Metode NWKNN dengan KNN

Pengujian perbandingan metode NWKNN dengan KNN dilakukan untuk mengetahui metode manakah yang menghasilkan tingkat akurasi tertinggi pada masing-masing nilai K yang diubah dengan rasio perbandingan jumlah data latih dan data uji yang sama. Pengujian dilakukan dengan mengubah nilai K sebesar 2 hingga nilai K sebesar 15 dengan nilai E yang digunakan ialah nilai E yang meghasilkan akurasi tertinggi pada pengujian sebelumnya untuk pengujian pada metode NWKNN. Rasio data latih dan data uji yang digunakan ialah rasio yang menghasilkan akurasi tertinggi pada pengujian sebelumnya. Gambaran Tabel pengujian perbandingan metode NWKNN dengan KNN ditunjukkan pada Tabel 4.14.

**Tabel 4.14 Tabel pengujian perbandingan metode NWKNN dengan KNN**

| Nilai K | Akurasi NWKNN | Akurasi KNN |
|---|---|---|
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |
| . | | |
| . | | |
| 15 | | |

# BAB 5 IMPLEMENTASI

Pada bab implementasi ini membahas implementasi sistem pengklasifikasian jenis kanker berdasarkan struktur protein menggunakan algoritma *Neighbor Weighted K-Nearest Neighbor*. Bab ini terdiri dari sub-bab spesifikasi sistem, batasan implementasi, implementasi *source code,* dan implementasi antarmuka.

## 5.1 Spesifikasi Sistem

Spesifikasi sistem yang digunakan dalam penelitian ini mengacu pada spesifikasi sistem yang digunakan oleh penulis untuk pengimplentasiannya. Spesifikasi sistem terbagi menjadi dua, yaitu spesifikasi perangkat keras dan spesifikasi perangkat lunak.

### 5.1.1 Spesifikasi Perangkat Keras

Implementasi sistem untuk klasifikasi jenis kanker berdasarkan struktur protein ini menggunakan laptop dengan spesifikasi perangkat keras sebagai berikut:

a. *Processor* Intel(R) Core(TM) i5-4200U CPU @ 1.60GHz 2.30GHz
b. Kapasitas Memori (RAM) sebesar 4.00 GB

### 5.1.2 Spesifikasi Perangkat Lunak

Implementasi sistem untuk klasifikasi jenis kanker berdasarkan struktur protein ini menggunakan laptop dengan spesifikasi perangkat lunak sebagai berikut:

a. Sistem Operasi Windows 10 Pro 64 bit
b. Bahasa Pemrograman Java
c. *Tools* pemrograman Netbeans 8.1

## 5.2 Batasan Implementasi

Batasan dalam implementasi sistem klasifikasi jenis kanker dengan menggunakan metode NWKNN adalah sebagai berikut:

1. Sistem dibuat menggunakan bahasa pemrograman java.
2. Data yang digunakan dalam implementasi sistem disimpan dalam bentuk teks dengan format data berekstensi txt.
3. Metode yang digunakan dalam identifikasi jenis kanker adalah metode NWKNN.
4. Input yang digunakan dalam sistem merupakan data latih, data uji, data *wild*, jumlah data latih, nilai K dan E.
5. Output yang dihasilkan ialah jenis kelas dari data uji
6. Ketiga data uji, data latih dan data *wild* harus sama sebanyak 393 karakter.

## 5.3 Implementasi Algoritma

Pada sub-bab implementasi algoritma akan dijelaskan tentang kode dari sistem klasifikasi jenis kanker berdasarkan struktur protein yang mengacu pada bab perancangan sub-bab perancangan proses yang meliputi proses perhitungan pada setiap langkah yang ada pada algoritma NWKNN.

### 5.3.1 Implementasi Pendefinisian Data

Terdapat *source code* untuk mendefinisikan data dalam program dengan cara membaca *file* berekstensi .txt menggunakan fungsi *BufferedReader*. Data yang dibaca setiap baris tersebut akan disimpan dalam array yang berbeda pada setiap jenis data (data sekuens, data kelas, data *wild*). Implementasi proses pendefinisian data terdapat pada *Source Code* 5.1.

```
1    // Membaca data
2                BufferedReader       bufread       =       new
3  BufferedReader(new InputStreamReader(System.in));
4
5                // Membaca dataset
6                FileReader       frSekuens       =       new
7  FileReader("D:\\DataSkripsi\\dataSekuens.txt");
8                Scanner scSekuens = new Scanner(frSekuens);
9                String[] dataSekuens = new String[752];
10               for (int i = 0; i < 752; i++) {
11                   if (scSekuens.hasNext()) {
12                       dataSekuens[i] = (scSekuens.next());
13                   }
14               }
15
16               // Membaca data kelas pada dataset
17               FileReader       frKelas       =       new
18 FileReader("D:\\DataSkripsi\\dataKelas.txt");
19               Scanner scKelas = new Scanner(frKelas);
20               String[] kelasData = new String[752];
21               for (int i = 0; i < 752; i++) {
22                   if (scKelas.hasNext()) {
23                       kelasData[i] = (scKelas.next());
24                   }
25               }
26
27               // Membaca data wild
28               FileReader       frWild       =       new
29 FileReader("D:\\DataSkripsi\\dataWild.txt");
30               Scanner scWild = new Scanner(frWild);
31               String dataWild;
32               dataWild = (scWild.next());
33            textareaWild.setText(dataWild);
```

***Source Code* 5.1 Implementasi Pendefinisian Data**

Berikut merupakan penjelasan *Source Code* 5.1 baris ke:

2-3 Membaca karakter pada file

6-7 Membaca file berekstensi txt yang berisi data sekuens protein pada dataset

8-14 Membaca semua isi file data sekuens dan menyimpannya ke dalam array bernama dataSekuens sebanyak 752 array

17-18 Membaca file berekstensi txt yang berisi data kelas pada dataset
19-25 Membaca semua isi file data kelas dan menyimpannya ke dalam array bernama kelasData sebanyak 752 array
28-29 Membaca file berekstensi txt yang berisi sekuns data *wild*
30-32 Membaca isi file data *wild* dan menyimpannya pada variabel dataWild yang telah diinisialisasi
33 Menampilkan data *wild* pada *text area* bernama textareaWild

### *5.3.2* Implementasi *Preprocessing*

Tahap *preprocessing* dilakukan untuk mengubah masing-masing data sekuens menjadi data numerik agar dapat dilakukan penghitungan menggunakan algoritma NWKNN. *Preprocessing* dilakukan dengan cara mencocokkan masing-masing karakter pada sekuens antara dataset dengan data *wild* berdasarkan tabel Matriks PAM250. Pada program ini dibuat array dua dimensi untuk membentuk matriks PAM250 tersebut. Implementasi *preprocessing* dapat dilihat pada *Source Code* 5.2.

```
1   // MEMBENTUK MATRIKS PAM
2               Double[][] tabelPAM = new Double[200][200];
3               tabelPAM['W']['W'] = 17.0;
4               tabelPAM['Y']['Y'] = 10.0;
5               tabelPAM['F']['F'] = 9.0;
6               tabelPAM['C']['C'] = 12.0;
7               tabelPAM['A']['A']    =   tabelPAM['A']['S']    =
8   tabelPAM['S']['G'] = tabelPAM['R']['W'] = tabelPAM['V']['L']
9   =     tabelPAM['D']['Q']      =     tabelPAM['I']['L']      =
10  tabelPAM['N']['N'] = tabelPAM['M']['I'] = tabelPAM['M']['V']
11  =     tabelPAM['Q']['E']      =     tabelPAM['L']['F']      =
12  tabelPAM['T']['T']      =     tabelPAM['F']['L']      =
13  tabelPAM['S']['A'] = tabelPAM['W']['R'] = tabelPAM['G']['S']
14  =     tabelPAM['Q']['D']      =     tabelPAM['S']['S']      =
15  tabelPAM['P']['S'] = tabelPAM['D']['N'] = tabelPAM['I']['M']
16  =     tabelPAM['N']['D']      =     tabelPAM['L']['I']      =
17  tabelPAM['N']['S'] = tabelPAM['E']['Q'] = tabelPAM['S']['N']
18  =     tabelPAM['V']['M']      =     tabelPAM['S']['P']      =
19  tabelPAM['L']['V'] = 2.0;
20              tabelPAM['L']['L']    =   tabelPAM['R']['R']    =
21  tabelPAM['P']['P'] = tabelPAM['H']['H'] = tabelPAM['M']['M']
22  = 6.0;
23              tabelPAM['M']['L']    =   tabelPAM['E']['E']    =
24  tabelPAM['L']['M'] = tabelPAM['I']['V'] = tabelPAM['D']['D']
25  =     tabelPAM['V']['V']      =     tabelPAM['Q']['Q']      =
26  tabelPAM['V']['I'] = tabelPAM['I']['I'] = 4.0;
27              tabelPAM['G']['G'] = tabelPAM['K']['K'] = 5.0;
28              tabelPAM['L']['A']    =   tabelPAM['A']['L']    =
29  tabelPAM['Q']['L'] = tabelPAM['L']['Q'] = tabelPAM['L']['T']
30  =     tabelPAM['T']['L']      =     tabelPAM['S']['L']      =
31  tabelPAM['L']['S'] = tabelPAM['A']['R'] = tabelPAM['R']['A']
32  =     tabelPAM['N']['I']      =     tabelPAM['I']['N']      =
33  tabelPAM['R']['I'] = tabelPAM['I']['R'] = tabelPAM['L']['H']
34  =     tabelPAM['H']['L']      =     tabelPAM['I']['K']      =
35  tabelPAM['K']['I'] = tabelPAM['L']['P'] = tabelPAM['P']['L']
36  =     tabelPAM['N']['Y']      =     tabelPAM['Y']['N']      =
37  tabelPAM['I']['P'] = tabelPAM['P']['I'] = tabelPAM['M']['P']
```

```
=     tabelPAM['P']['M']     =     tabelPAM['H']['M']     =
tabelPAM['M']['H'] = tabelPAM['E']['I'] = tabelPAM['I']['E']
=     tabelPAM['N']['M']     =     tabelPAM['M']['N']     =
tabelPAM['H']['T'] = tabelPAM['T']['H'] = tabelPAM['F']['S']
=     tabelPAM['S']['F']     =     tabelPAM['G']['K']     =
tabelPAM['K']['G'] = tabelPAM['A']['C'] = tabelPAM['C']['A']
=     tabelPAM['Q']['I']     =     tabelPAM['I']['Q']     =
tabelPAM['I']['C'] = tabelPAM['C']['I'] = tabelPAM['E']['M']
=     tabelPAM['M']['E']     =     tabelPAM['S']['W']     =
tabelPAM['W']['S'] = tabelPAM['H']['F'] = tabelPAM['F']['H']
=     tabelPAM['D']['I']     =     tabelPAM['I']['D']     =
tabelPAM['M']['Y'] = tabelPAM['Y']['M'] = tabelPAM['T']['Q']
=     tabelPAM['Q']['T']     =     tabelPAM['R']['T']     =
tabelPAM['T']['R'] = -2.0;
            tabelPAM['Q']['V']   =   tabelPAM['V']['Q']   =
tabelPAM['R']['V'] = tabelPAM['V']['R'] = tabelPAM['K']['V']
=     tabelPAM['V']['K']     =     tabelPAM['C']['V']     =
tabelPAM['V']['C'] = tabelPAM['H']['V'] = tabelPAM['V']['H']
=     tabelPAM['E']['V']     =     tabelPAM['V']['E']     =
tabelPAM['G']['V'] = tabelPAM['V']['G'] = tabelPAM['N']['V']
=     tabelPAM['V']['N']     =     tabelPAM['D']['V']     =
tabelPAM['V']['D'] = -2.0;
            tabelPAM['G']['P']   =   tabelPAM['P']['G']   =
tabelPAM['E']['H'] = tabelPAM['H']['E'] = tabelPAM['R']['N']
=     tabelPAM['N']['R']     =     tabelPAM['E']['A']     =
tabelPAM['A']['E'] = tabelPAM['D']['K'] = tabelPAM['D']['K']
=     tabelPAM['N']['G']     =     tabelPAM['G']['N']     =
tabelPAM['T']['V'] = tabelPAM['V']['T'] = tabelPAM['N']['T']
=     tabelPAM['T']['N']     =     tabelPAM['H']['S']     =
tabelPAM['S']['H'] = tabelPAM['E']['G'] = tabelPAM['G']['E']
=     tabelPAM['D']['P']     =     tabelPAM['P']['D']     =
tabelPAM['M']['F'] = tabelPAM['F']['M'] = tabelPAM['A']['T']
=     tabelPAM['T']['A']     =     tabelPAM['P']['Q']     =
tabelPAM['Q']['P'] = tabelPAM['P']['R'] = tabelPAM['R']['P']
=     tabelPAM['E']['P']     =     tabelPAM['P']['E']     =
tabelPAM['H']['K'] = tabelPAM['K']['H'] = tabelPAM['A']['N']
=     tabelPAM['N']['A']     =     tabelPAM['K']['E']     =
tabelPAM['E']['K'] = tabelPAM['A']['D'] = tabelPAM['D']['A']
=     tabelPAM['A']['V']     =     tabelPAM['V']['A']     =
tabelPAM['S']['Q'] = tabelPAM['Q']['S'] = tabelPAM['H']['P']
=     tabelPAM['P']['H']     =     tabelPAM['Q']['A']     =
tabelPAM['A']['Q'] = tabelPAM['P']['T'] = tabelPAM['T']['P']
=     tabelPAM['W']['Y']     =     tabelPAM['Y']['W']     =
tabelPAM['N']['P'] = tabelPAM['P']['N'] = tabelPAM['H']['Y']
= tabelPAM['Y']['H'] = 0.0;
            tabelPAM['K']['A']   =   tabelPAM['A']['K']   =
tabelPAM['R']['D'] = tabelPAM['D']['R'] = tabelPAM['K']['P']
=     tabelPAM['P']['K']     =     tabelPAM['L']['Y']     =
tabelPAM['Y']['L'] = tabelPAM['E']['T'] = tabelPAM['T']['E']
=     tabelPAM['A']['M']     =     tabelPAM['M']['A']     =
tabelPAM['M']['S'] = tabelPAM['S']['M'] = tabelPAM['I']['T']
=     tabelPAM['T']['I']     =     tabelPAM['K']['T']     =
tabelPAM['T']['K'] = tabelPAM['P']['V'] = tabelPAM['V']['P']
=     tabelPAM['M']['Q']     =     tabelPAM['Q']['M']     =
tabelPAM['T']['D'] = tabelPAM['D']['T'] = tabelPAM['P']['C']
=     tabelPAM['C']['P']     =     tabelPAM['I']['A']     =
tabelPAM['A']['I'] = tabelPAM['R']['M'] = tabelPAM['M']['R']
=     tabelPAM['M']['T']     =     tabelPAM['T']['M']     =
tabelPAM['G']['T'] = tabelPAM['T']['G'] = tabelPAM['S']['V']
```

```
97  =     tabelPAM['V']['S']     =     tabelPAM['I']['Y']     =
98  tabelPAM['Y']['I'] = tabelPAM['A']['H'] = tabelPAM['H']['A']
99  =     tabelPAM['F']['V']     =     tabelPAM['V']['F']     =
100 tabelPAM['I']['S'] = tabelPAM['S']['I'] = tabelPAM['G']['Q']
101 = tabelPAM['Q']['G'] = tabelPAM['D']['C'] = -1.0;
102             tabelPAM['N']['L']   =   tabelPAM['L']['N']   =
103 tabelPAM['H']['I'] = tabelPAM['I']['H'] = tabelPAM['G']['R']
104 =     tabelPAM['R']['G']     =     tabelPAM['P']['C']     =
105 tabelPAM['C']['P'] = tabelPAM['R']['L'] = tabelPAM['L']['R']
106 =     tabelPAM['A']['F']     =     tabelPAM['F']['A']     =
107 tabelPAM['C']['R'] = tabelPAM['R']['C'] = tabelPAM['A']['Y']
108 =     tabelPAM['Y']['A']     =     tabelPAM['M']['D']     =
109 tabelPAM['D']['M'] = tabelPAM['D']['F'] = tabelPAM['F']['D']
110 =     tabelPAM['L']['K']     =     tabelPAM['K']['L']     =
111 tabelPAM['H']['C'] = tabelPAM['C']['H'] = tabelPAM['G']['I']
112 =     tabelPAM['I']['G']     =     tabelPAM['S']['Y']     =
113 tabelPAM['Y']['S'] = tabelPAM['G']['M'] = tabelPAM['M']['G']
114 =     tabelPAM['H']['G']     =     tabelPAM['G']['H']     =
115 tabelPAM['C']['T'] = tabelPAM['T']['C'] = tabelPAM['L']['E']
116 =     tabelPAM['E']['L']     =     tabelPAM['G']['C']     =
117 tabelPAM['C']['G'] = tabelPAM['Y']['V'] = tabelPAM['V']['Y']
118 = tabelPAM['T']['Y'] = tabelPAM['Y']['T'] = -3.0;
119             tabelPAM['E']['Y']   =   tabelPAM['Y']['E']   =
120 tabelPAM['R']['F'] = tabelPAM['F']['R'] = tabelPAM['F']['Q']
121 =     tabelPAM['Q']['F']     =     tabelPAM['N']['C']     =
122 tabelPAM['C']['N'] = tabelPAM['F']['P'] = tabelPAM['P']['F']
123 =     tabelPAM['F']['T']     =     tabelPAM['T']['F']     =
124 tabelPAM['G']['F'] = tabelPAM['F']['G'] = tabelPAM['D']['Y']
125 =     tabelPAM['Y']['D']     =     tabelPAM['K']['W']     =
126 tabelPAM['W']['K'] = tabelPAM['G']['L'] = tabelPAM['L']['G']
127 =     tabelPAM['Y']['Q']     =     tabelPAM['Q']['Y']     =
128 tabelPAM['L']['D'] = tabelPAM['D']['L'] = tabelPAM['C']['F']
129 = tabelPAM['F']['C'] = -4.0;
130             tabelPAM['E']['F']   =   tabelPAM['F']['E']   =
131 tabelPAM['C']['K'] = tabelPAM['K']['C'] = tabelPAM['Q']['C']
132 =     tabelPAM['C']['Q']     =     tabelPAM['K']['F']     =
133 tabelPAM['F']['K'] = tabelPAM['G']['Y'] = tabelPAM['Y']['G']
134 =     tabelPAM['P']['Y']     =     tabelPAM['Y']['P']     =
135 tabelPAM['T']['W'] = tabelPAM['W']['T'] = tabelPAM['D']['C']
136 =     tabelPAM['C']['D']     =     tabelPAM['C']['M']     =
137 tabelPAM['M']['C'] = tabelPAM['R']['Y'] = tabelPAM['Y']['R']
138 =     tabelPAM['N']['W']     =     tabelPAM['W']['N']     =
139 tabelPAM['D']['F'] = tabelPAM['F']['D'] = tabelPAM['K']['Y']
140 =     tabelPAM['Y']['K']     =     tabelPAM['H']['W']     =
141 tabelPAM['W']['H'] = tabelPAM['E']['C'] = tabelPAM['C']['E']
142 = -5.0;
143             tabelPAM['N']['K']   =   tabelPAM['K']['N']   =
144 tabelPAM['G']['A'] = tabelPAM['A']['G'] = tabelPAM['H']['N']
145 =     tabelPAM['N']['H']     =     tabelPAM['A']['P']     =
146 tabelPAM['P']['A'] = tabelPAM['D']['H'] = tabelPAM['H']['D']
147 =     tabelPAM['N']['E']     =     tabelPAM['E']['N']     =
148 tabelPAM['R']['E'] = tabelPAM['E']['R'] = tabelPAM['I']['F']
149 =     tabelPAM['F']['I']     =     tabelPAM['R']['S']     =
150 tabelPAM['S']['R'] = tabelPAM['K']['Q'] = tabelPAM['Q']['K']
151 =     tabelPAM['K']['M']     =     tabelPAM['M']['K']     =
152 tabelPAM['R']['Q'] = tabelPAM['Q']['R'] = tabelPAM['R']['H']
153 =     tabelPAM['H']['R']     =     tabelPAM['C']['S']     =
154 tabelPAM['S']['C'] = tabelPAM['K']['S'] = tabelPAM['S']['K']
155 =     tabelPAM['E']['S']     =     tabelPAM['S']['E']     =
```

```
tabelPAM['D']['G'] = tabelPAM['G']['D'] = tabelPAM['F']['W'] =     tabelPAM['W']['F']     =     tabelPAM['N']['Q']     = tabelPAM['Q']['N'] = tabelPAM['S']['T'] = tabelPAM['T']['S'] =     tabelPAM['C']['Y']     =     tabelPAM['Y']['C']     = tabelPAM['D']['S'] = tabelPAM['S']['D'] = tabelPAM['Q']['I'] = 1.0;
            tabelPAM['R']['K']   =   tabelPAM['K']['R']   = tabelPAM['H']['Q'] = tabelPAM['Q']['H'] = tabelPAM['E']['D'] = tabelPAM['D']['E'] = 3.0;
            tabelPAM['W']['P']   =   tabelPAM['P']['W']   = tabelPAM['I']['W'] = tabelPAM['W']['I'] = tabelPAM['A']['W'] =     tabelPAM['W']['A']     =     tabelPAM['C']['L']     = tabelPAM['L']['C'] = tabelPAM['W']['V'] = tabelPAM['V']['W'] =     tabelPAM['W']['Q']     =     tabelPAM['Q']['W']     = tabelPAM['M']['W'] = tabelPAM['W']['M'] = -6.0;
            tabelPAM['C']['W']   =   tabelPAM['W']['C']   = tabelPAM['E']['W'] = tabelPAM['W']['E'] = tabelPAM['L']['W'] =     tabelPAM['W']['L']     =     tabelPAM['D']['W']     = tabelPAM['W']['D'] = tabelPAM['G']['W'] = tabelPAM['W']['G'] = -7.0;
            tabelPAM['Y']['F'] = tabelPAM['F']['Y'] =  7.0;

            // Konversi data latih
            Double[][]        konvLatih         =         new Double[dataLatih.length][393];
            String[]        stringKonvLatih       =       new String[konvLatih.length];
            for (int i = 0; i < konvLatih.length; i++) {
                String str = "";
                for (int j = 0; j < konvLatih[0].length; j++) {
                    String latih = dataLatih[i];
                    konvLatih[i][j]                         = tabelPAM[latih.charAt(j)][dataWild.charAt(j)];
                    str += String.valueOf(konvLatih[i][j]) + "\t";
                }
                stringKonvLatih[i] = str;
            }

            // Konversi data uji
            Double[][]        konvUji         =         new Double[dataUji.length][393];
            String[]        stringKonvUji       =       new String[konvUji.length];
            for (int i = 0; i < konvUji.length; i++) {
                String str = "";
                for (int j = 0; j < konvUji[0].length; j++) {
                    String uji = dataUji[i];
                    konvUji[i][j]                         = tabelPAM[uji.charAt(j)][dataWild.charAt(j)];
                    str  +=  String.valueOf(konvUji[i][j])  + "\t";
                }
                stringKonvUji[i] = str;}
```

***Source Code* 5.2 Implementasi *Preprocessing***

Berikut merupakan penjelasan *Source Code* 5.2 baris ke:

1 Inisialisasi array PAM 2 dimensi

2-176 Inisialisasi masing-masing array sebagai tabel matriks PAM250

179 Inisialisasi array konvLatih 2 dimensi dengan baris sebanyak jumlah data latih dan panjang sepanjang 393. 393 merupakan jumlah karakter pada data sekuens

181 Inisialisasi array stringKonvLatih 1 dimensi dengan baris sebanyak baris konvLatih. Array ini digunakan untuk mencetak tampilan hasil *preprocessing*

183 Melakukan perulangan dengan menggunakan *for looping* sebanyak baris konvLatih

184 Memisahkan masing-masing karakter dengan “”

185-194 Melakukan perulangan dengan menggunakan *for looping* sebanyak panjang konvLatih dan mencocokkan antara karakter data latih dengan data *wild* pada array PAM dan nilainya disimpan pada array konvLatih

197 Inisialisasi array konvUji 2 dimensi dengan baris sebanyak jumlah data uji dan panjang sepanjang 393. 393 merupakan jumlah karakter pada data sekuens

199 Inisialisasi array stringKonvUji 1 dimensi dengan baris sebanyak baris konvUji. Array ini digunakan untuk mencetak tampilan hasil *preprocessing*

203-213 Melakukan perulangan dengan menggunakan *for looping* sebanyak panjang konvUji dan mencocokkan antara karakter data latih dengan data *wild* pada array PAM dan nilainya disimpan pada array konvUji

### 5.3.3 Implementasi Menghitung Nilai Kedekatan Ketetanggaan dengan CosSim

Tahap menghitung nilai kedekatan ketetanggaan antara data latih dengan data uji dihitung dengan menggunakan Persamaan 2.1. Pada program ini penghitungan dibagi menjadi beberapa tahap untuk memudahkan pengimplementasian. Implementasi menghitung nilai kedekatan terdapat pada *Source Code* 5.3.

```
1  Double[] ketetanggaan = new Double[konvLatih.length];
2                  for (int i = 0; i < konvLatih.length; i++) {
3                      double totalPembilang = 0.0;
4                      double totalPenyebutLatih = 0.0;
5                      double totalPenyebutUji = 0.0;
6                      for (int j = 0; j < konvLatih[0].length;
7  j++) {
8                          totalPembilang += (konvUji[x][j] *
9  konvLatih[i][j]);
10                         totalPenyebutLatih                 +=
11 Math.pow(konvLatih[i][j], 2);
12                         totalPenyebutUji                   +=
13 Math.pow(konvUji[x][j], 2);
14                     }
15 double        hasil        =        totalPembilang        /
16 (Math.sqrt(totalPenyebutLatih)                             *
17 Math.sqrt(totalPenyebutUji));
18                     ketetanggaan[i] = hasil;
```

```
19                      hasilJarak[x][i] = ketetanggaan[i];
20                  }
```

***Source Code* 5.3 Implementasi Menghitung Nilai Kedekatan Ketetanggaan**

Berikut merupakan penjelasan *Source Code* 5.3 baris ke:

1 Inisialisasi variabel array ketetanggaan dengan baris data sebanyak baris konvLatih
2 Melakukan perulangan sebanyak baris konvLatih
3-5 Inisialisasi masing-masing variabel totalPembilang, totalPenyebutLatih dan totalPenyebutUji
6-7 Melakukan perulangan sebanyak panjang konvLatih
8-9 Melakukan penjumlahan pada penghitungan variabel totalPembilang yang didapat dari hasil perkalian konvUji dan konvLatih
10-11 Melakukan penjumlahan pada penghitungan variabel totalPenyebutLatih yang didapat dari kuadrat konvLatih
12-13 Melakukan penjumlahan pada penghitungan variabel totalPenyebutUji yang didapat dari kuadrat konvUji
15-17 Inisialisasi variabel hasil dengan membagi antara totalPembilang dengan perkalian dari akar totalPenyebutLatih dan totalPenyebutUji
18 Inisialisasi ketetanggaan dengan nilai hasil
19 Inisialisasi hasilJarak dengan ketetanggaan

### 5.3.4 Implementasi Mengurutkan Data dengan Kedeketan Ketetanggaan Terbesar hingga Terkecil

Pada tahap mengurutkan data ini, program menggunakan fungsi *clone* untuk duplikasi dataKelas agar tidak terjadi perubahan posisi dataKelas pada saat penghitungan dengan data uji selanjutnya. Implementasi pengurutan terdapat pada *Source Code* 5.4.

```
1  String[] dataKelasBaru = dataKelas.clone();
2                  // Mengurutkan nilai ketetanggaan dari
3  ketetanggaan terbesar ke terkecil
4                  for (int i = 0; i < ketetanggaan.length; i++)
5  {
6                      for (int j = 0; j < ketetanggaan.length -
7  1; j++) {
8                          if (ketetanggaan[j] < ketetanggaan[j
9  + 1]) {
10                             double temp = ketetanggaan[j];
11                             String temp1 = dataKelasBaru[j];
12                             ketetanggaan[j] = ketetanggaan[j
13 + 1];
14                             dataKelasBaru[j]                =
15 dataKelasBaru[j + 1];
16                             ketetanggaan[j + 1] = temp;
17                             dataKelasBaru[j + 1] = temp1;
18                         }
19                     }
20                 }
21
22 for (int i = 0; i < ketetanggaan.length; i++) {
23                     ketetanggaanTerurut[x][i]               =
24 ketetanggaan[i];
```

```
25                      kelasTerurut[x][i] = dataKelasBaru[i];
26                  }
```

***Source Code* 5.4 Implementasi Mengurutkan Data dengan Kedekatan Ketetanggaan Terbesar hingga Terkecil**

Berikut merupakan penjelasan *Source Code* 5.4 baris ke:

1 Duplikasi dataKelas ke variabel dataKelasBaru agar urutan asli dari dataKelas tidak berubah

4-5 Melakukan perulangan sebanyak baris ketetanggaan

6-7 Melakukan perulangan sebanyak baris ketetanggaan-1

8-20 Melakukan pengecekan kondisi apakah ketetanggaan [j] lebih kecil dibanding ketetanggaan[j+1]. Apabila kondisi terpenuhi, maka akan terjadi pemindahan data ketetanggaan dan dataKelasBaru

22-26 Inisialisasi ketetanggaanTerurut dengan ketetanggaan dan kelasTerurut dengan dataKelasBaru

### 5.3.5 Implementasi Pembobotan Setiap Kelas

Sebelum dilakukan penghitungan nilai skor, dilakukan penghitungan bobot pada tiap kelas terlebih dahulu. Penghitungan bobot ini perlu diketahui beberapa hal, seperti mengetahui banyaknya data pada setiap jumlah kelas serta jumlah minimal banyaknya data pada dataset. Implementasi pembobotan setiap kelas ini terdapat pada *Source Code* 5.5.

```
1   // Menghitung jumlah kelas
2               double[] jmlKelas = new double[4];
3               for(int i = 0; i < jml_datalatih; i++){
4                   if(kelasData[i].equals("0")){
5                       jmlKelas[0] += 1;
6                   }
7                   else if(kelasData[i].equals("1")){
8                       jmlKelas[1] += 1;
9                   }
10                  else if(kelasData[i].equals("2")){
11                      jmlKelas[2] += 1;
12                  }
13                  else{
14                      jmlKelas[3] += 1;
15                  }
16              }
17
18              // Mencari kelas dengan data minimal
19              double jmlKelasMin = 0;
20              double tempMin = jmlKelas[0];
21              for(int i = 1; i < jmlKelas.length-1; i++){
22                  if(jmlKelas[i] < tempMin){
23                      jmlKelasMin = jmlKelas[i];
24                      tempMin = jmlKelasMin;
25                  }
26              }
27
28              // Menghitung Bobot masing-masing kelas
29              double [] bobotKelas = new double[4];
30              double penyebutAkar = 1/exp;
31              for(int i = 0; i< bobotKelas.length; i++){
```

```
32                  bobotKelas[i]          =          1          /
33  Math.pow((jmlKelas[i]/jmlKelasMin), penyebutAkar);
34                  textareaBobot.append("Bobot  Kelas-"+i+"  ==
35  "+bobotKelas[i]+"\n");
36              }
```

***Source Code* 5.5 Implementasi Pembobotan Setiap Kelas**

Berikut merupakan penjelasan *Source Code* 5.5 baris ke:

2 Inisialisasi array jmlKelas sebanyak 4 baris bertipe data Double

3-16 Perulangan sebanyak jml_dataLatih dengan pengecekan kondisi apabila kelasData cocok dengan masing-masing jenis kelasnya (0, 1, 2, 3), maka akan ditambah 1 pada kelas yang cocok

19 Inisialisasi jmlKelasMin dengan 0 bertipe data Double

20 Inisialisasi tempMin dengan jmlKelas[0] untuk menampung nilai sementara sebagai acuan apakah nilai yang dibandingkan bernilai lebih besar atau tidak dengan tipe data Double

21-26 Perulangan sebanyak jumlah baris jmlKelas – 1 untuk memindahkan jmlKelas agar mendapatkan nilai terkecil

29 Inisialisasi array bobotKelas sebanyak 4 baris bertipe data Double

30 Inisialisasi penyebutAkar dengan 1/exp bertipe data Double. exp merupakan nilai *Exponent* yang awalnya sudah diinsialisasi oleh masukan user

31-33 Perulangan sebanyak baris bobotKelas dan melakukan inisialisasi terhadap bobotKelas dengan membagi 1 dengan hasil dari pengakaran pembagian antara jmlKelas dan jmlKelasMin lalu ditampilkan pada *text area textareaBobot*

### 5.3.6 Implementasi Pengambilan Data Sebanyak K

Pengambilan data sebanyak K dilakukan untuk mengambil beberapa kelas data sebagai nilai berdasarkan tetangganya untuk klasifikasi. Nilai K menjadi nilai jumlah banyaknya data yang diambil sebagai kelas data untuk klasifikasi. Implementasi pengambilan data sebanyak K ini terdapat pada *Source Code* 5.6.

```
1  for (int i = 0; i < k; i++) {
2                      kelask[x][i] = kelasTerurut[x][i];
3                      kketetanggaan[x][i]                    =
4  ketetanggaanTerurut[x][i];
5                  }
6
```

***Source Code* 5.6 Implementasi Pengambilan Data Sebanyak K**

Berikut merupakan penjelasan *Source Code* 5.6 baris ke:

1 Melakukan perulangan sebanyak nilai K yang telah didefinisikan sebelumnya

2 Inisialisasi kelask dengan kelasTerurut

4-5 Inisialisasi kketetanggaan dengan ketetanggaanTerurut

### 5.3.7 Implementasi Penghitungan Nilai Skor

Penghitungan nilai skor ini diperlukan untuk mengetahui nilai skor yang didapat pada masing-masing kelas untuk digunakan sebagai perbandingan klasifikasi. Pada implementasi ini dibuat array pada masing-masing kelasnya untuk menjumlahkan skor sementara, yaitu skor yang hanya berisi jumlah masing-masing nilai ketetanggaannya dan belum dikalikan dengan bobot kelas. Setelah didapatkan skor sementara, skor sementara tersebut dikalikan dengan masing-masing bobot kelas dan dimasukkan di array yang berbeda. Implementasi penghitungan nilai skor ini terdapat pada *Source Code* 5.7.

```
1   double[] tempSkor = new double [4];
2                   String setKelass[] = {"0", "1", "2", "3"};
3                   for (int i = 0; i < k; i++) {
4
5   if(kelasTerurut[x][i].equals(setKelass[0])){
6                           tempSkor[0]                         +=
7   ketetanggaanTerurut[x][i];
8
9                       }
10                      else
11  if(kelasTerurut[x][i].equals(setKelass[1])){
12                          tempSkor[1]                         +=
13  ketetanggaanTerurut[x][i];
14
15                      }
16                      else
17  if(kelasTerurut[x][i].equals(setKelass[2])){
18                          tempSkor[2]                         +=
19  ketetanggaanTerurut[x][i];
20
21                      }
22                      else{
23                          tempSkor[3]                         +=
24  ketetanggaanTerurut[x][i];
25
26                      }
27                  }
28
29                  for (int i = 0; i < 4; i++) {
30                      skorKelas[x][i]    =    tempSkor[i]     *
31  bobotKelas[i];
32                  }
```

***Source Code* 5.7 Implementasi Penghitungan Nilai Skor**

Berikut merupakan penjelasan *Source Code* 5.7 baris ke:

1 Inisialisasi tempSkor sebanyak 4 baris bertipe data Double

2 Inisialisasi array setKelass bertipe data String dengan masing-masing array memiliki data “0”, “1”, “2”, “3”

3 Melakukan perulangan sebanyak nilai K

5-9 Melakukan pengecekan kondisi apabila kelasTerurut berisi “0”, maka array tempSkor[0] akan dijumlahkan dengan ketetanggaanTerurut sebagai nilai skor sementara kelas 0 (*non-cancer*)

10-15 Melakukan pengecekan kondisi apabila kelasTerurut berisi “1”, maka array tempSkor[1] akan dijumlahkan dengan ketetanggaanTerurut sebagai nilai skor sementara kelas 1 (*breast cancer*)

16-21 Melakukan pengecekan kondisi apabila kelasTerurut berisi “2”, maka array tempSkor[2] akan dijumlahkan dengan ketetanggaanTerurut sebagai nilai skor sementara kelas 2 (*colorectal cancer*)

22-26 Melakukan pengecekan kondisi apabila kelasTerurut berisi “3”, maka array tempSkor[3] akan dijumlahkan dengan ketetanggaanTerurut sebagai nilai skor sementara kelas 3 (*lung cancer*)

29-32 Melakukan perulangan sebanyak 4 kali (jumlah jenis kanker yang ada di dataset) untuk inisialisasi skorKelas dengan mengalikan antara tempSkor dan bobotKelas sebagai skor final pada masing-masing kelas

### 5.3.8 Implementasi Komputasi Kelas Data Uji

Komputasi kelas data uji dilakukan untuk mengetahui klasifikasi kelas kanker pada data uji berdasarkan penghitungan menggunakan metode NWKNN. Pengurutan data ini menggunakan metode *bubble sort* untuk memindahkan data untuk mendapatkan skor kelas terbesar. Implementasi komputasi kelas data uji ini terdapat pada *Source Code* 5.8.

```
1  for (int i = 0; i < 4 ; i++) {
2                     klasifikasiKelas[x][i]                    =
3  Integer.toString(i);
4                 }
5
6                 //HASIL KLASIFIKASI
7                 for (int i = 0; i < 4; i++) {
8                     for (int j = 0; j < 3; j++) {
9                         if (skorKelas[x][j] < skorKelas[x][j
10 + 1]) {
11                            double temp = skorKelas[x][j];
12                            String          temp1          =
13 klasifikasiKelas[x][j];
14                            skorKelas[x][j] = skorKelas[x][j
15 + 1];
16                            klasifikasiKelas[x][j]         =
17 klasifikasiKelas[x][j + 1];
18                            skorKelas[x][j + 1] = temp;
19                            klasifikasiKelas[x][j   +   1]   =
20 temp1;
21                        }
22                    }
23                }
24
25                Uklasifikasi[x]                             =
26 String.valueOf(skorKelas[x][0]);
27                kelasKlasifikasi[x]                         =
28 klasifikasiKelas[x][0];
```

***Source Code* 5.8 Implementasi Komputasi Kelas Data Uji**

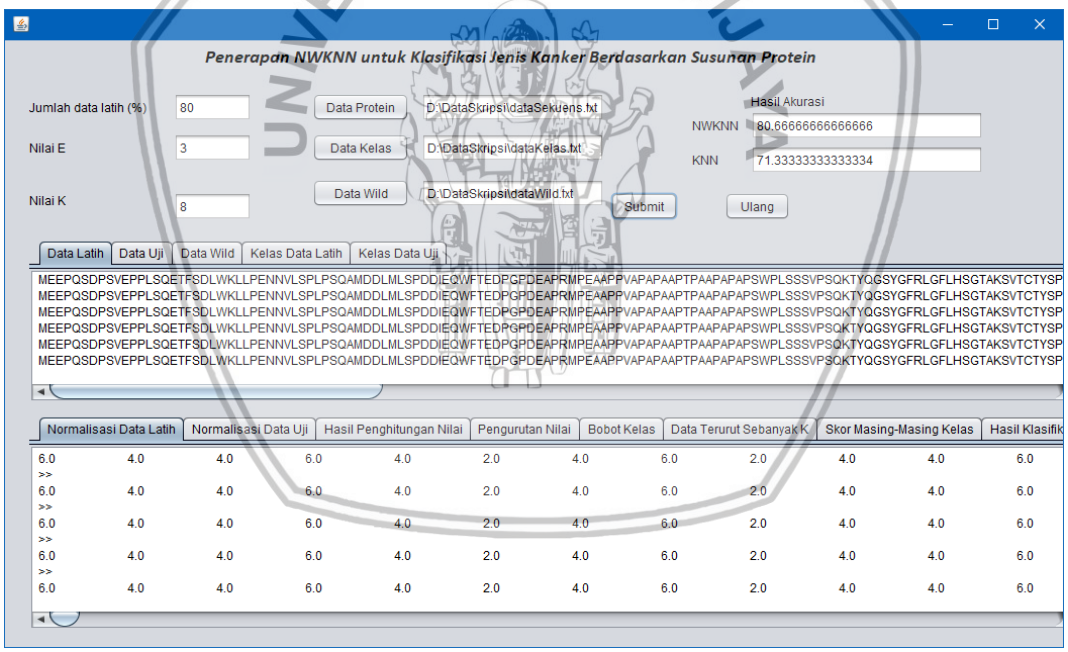
Berikut merupakan penjelasan *Source Code* 5.8 baris ke:

1-4 Melakukan perulangan sebanyak 4 kali untuk inisialisasi klasifikasiKelas dengan nilai i sebagai parameter kelas yang tersedia

7 Melakukan perulangan sebanyak 4 kali
8 Melakukan perulangan sebanyak 3 kali
9-21 Melakukan pengecekan kondisi apakah skorKelas[x][j] lebih kecil dibanding skorKelas[x][j+1]. Apabila kondisi terpenuhi, maka akan terjadi pemindahan data skorKelas dan klasifikasiKelas
25-26 Inisialisasi Uklasifikasi dengan skorKelas
27-28 Inisilasiasi kelasKlasifikasi dengan kelasKlasifikasi

## 5.4 Implementasi Antarmuka

Antarmuka atau *interface* sistem yang dibuat untuk klasifikasi jenis kanker dengan menggunakan metode NWKNN berdasarkan struktur protein ini digunakan oleh pengguna untuk melakukan klasifikasi. Pada antarmuka ini disajikan halaman antarmuka untuk menginput nilai E, nilai K, jumlah persen data latih dan akan ditampilkan berupa hasil akurasi, dataset dari data latih dan data uji, data *wild*, kelas data latih, kelas data uji, normalisasi data latih, normalisasi data uji, hasil penghitungan nilai, pengurutan nilai, bobot kelas, data terurut sebanyak K, skor masing-masing kelas, hasil klasifikasi. Implementasi antarmuka terdapat pada Gambar 5.1.




**Gambar 5.1 Implementasi antarmuka**

# BAB 6 PENGUJIAN DAN ANALISIS

## 6.1 Pengujian dan Analisis Pengaruh Perubahan Jumlah Data Latih dan Data Uji

Pengujian pengaruh perubahan jumlah data latih dan data uji dilakukan untuk mengetahui apakah perubahan terhadap perbandingan antara jumlah data latih dengan uji dapat berpengaruh pada tingkat akurasi. Perbandingan jumlah data latih dengan data uji yang digunakan ialah 90%:10%, 80%:20%, 70%:30%, 60%:40%, 50%:50%, 40%:60%, 30%:70%, 20%:80% dan 10%:90% dari dataset yang digunakan, menggunakan nilai E sebesar 3 dan menggunakan tiga nilai K, yaitu K=5, K=10 dan K=15. Detail jumlah data yang digunakan ditunjukkan pada Tabel 6.1.

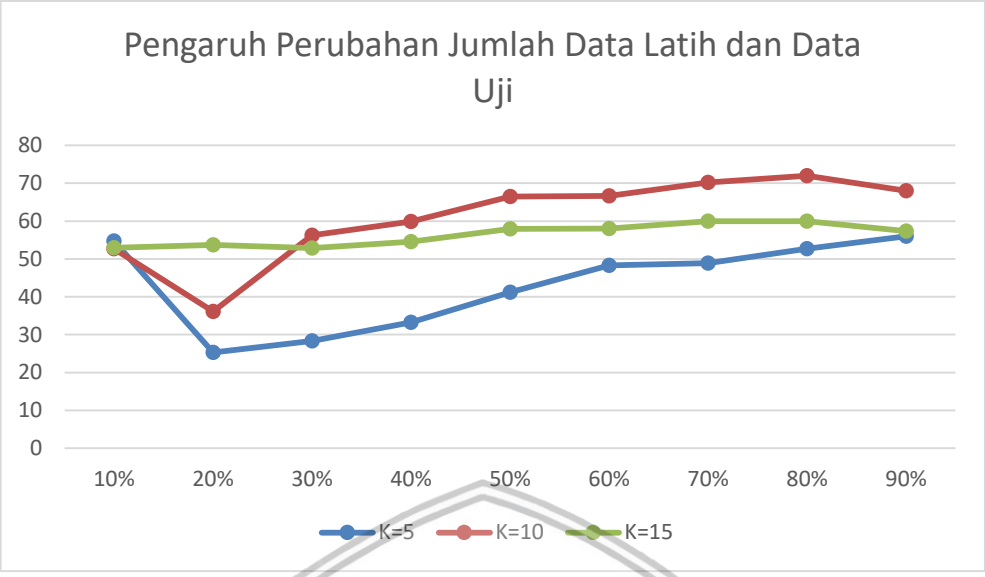
**Tabel 6.1 Jumlah Data yang Digunakan Pada Masing-Masing Rasio**

| Kelas yang Digunakan | Jumlah Data | | | |
|---|---|---|---|---|
| | Non-cancer (kelas 0) | Kanker Payudara (kelas 1) | Kanker Usus (kelas 2) | Kanker Paru-Paru (kelas 3) |
| 90%:10% | 357 | 103 | 107 | 110 |
| 80%:20% | 311 | 96 | 97 | 98 |
| 70%:30% | 267 | 85 | 85 | 90 |
| 60%:40% | 225 | 83 | 68 | 76 |
| 50%:50% | 179 | 68 | 62 | 67 |
| 40%:60% | 152 | 48 | 50 | 51 |
| 30%:70% | 118 | 33 | 40 | 35 |
| 20%:80% | 80 | 30 | 20 | 21 |
| 10%:90% | 41 | 17 | 9 | 9 |

Hasil pengujian pengaruh data latih dan data uji ditampilkan pada Tabel 6.1 dan grafik hasil pengujian ditampilkan pada Gambar 6.2.

**Tabel 6.2 Hasil pengujian pengaruh perubahan jumlah data latih dan data uji**

| Nilai E | Persentase Data Latih | Persentase Data Uji | Jumlah Data Latih | Jumlah Data Uji | Akurasi (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | K=5 | K=10 | K=15 | Rata-Rata |
| 3 | 90 | 10 | 677 | 75 | 56.000 | 68.000 | 57.333 | 60.444 |
| | 80 | 20 | 602 | 150 | 52.666 | 72.000 | 60.000 | 61.555 |
| | 70 | 30 | 527 | 225 | 48.888 | 70.222 | 60.000 | 59.703 |
| | 60 | 40 | 452 | 300 | 48.333 | 66.666 | 58.000 | 57.666 |
| | 50 | 50 | 376 | 376 | 41.223 | 66.489 | 57.978 | 55.230 |
| | 40 | 60 | 301 | 451 | 33.259 | 59.866 | 54.545 | 49.223 |
| | 30 | 70 | 226 | 526 | 28.326 | 56.273 | 52.851 | 45.816 |
| | 20 | 80 | 151 | 601 | 25.291 | 36.106 | 53.743 | 38.388 |
| | 10 | 90 | 76 | 676 | 54.733 | 52.662 | 52.958 | 53.451 |




**Gambar 6.1 Grafik pengaruh perubahan jumlah data latih dan data uji**

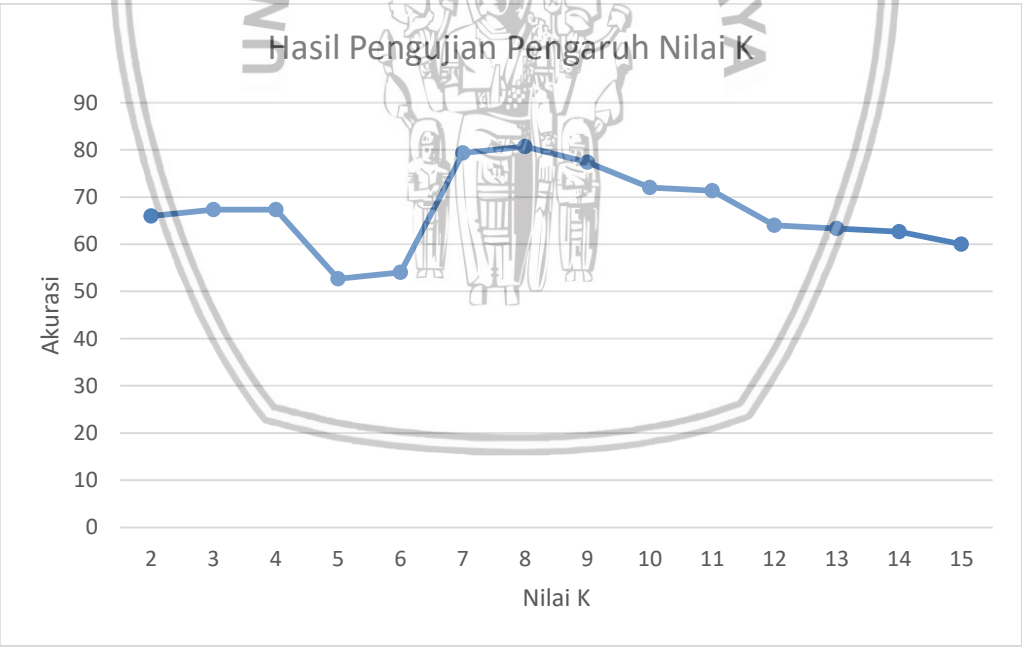
Berdasarkan Tabel 6.1 dan Gambar 6.1 dapat dilihat bahwa pengujian pengaruh perubahan data latih dan data uji menghasilkan nilai akurasi yang beragam. Grafik menunjukkan bahwa semakin banyak data latih yang digunakan, semakin baik nilai akurasi yang dihasilkan. Akurasi tertinggi dihasilkan pada saat pengujian menggunakan data latih sebanyak 80% dan akurasi terkecil dihasilkan pada saat pengujian menggunakan data latih sebesar 50%. Hal ini disebabkan karena saat semakin banyak data latih yang digunakan, maka semakin banyak pula data yang dibandingkan dengan masing-masing data uji sehingga nilai *CosSim* yang didapat merupakan data yang memiliki *similarity* yang mendekati data uji yang diklasifikasi cenderung tinggi untuk masuk ke dalam ketetanggaannya sehingga sistem mampu mengenali data yang lebih beragam yang dijadikan sebagai pembelajaran oleh sistem. Sebaliknya, apabila data latih yang digunakan sedikit, maka semakin sedikit pula data yang dibandingkan dengan data uji karena sedikitnya data yang beragam sebagai pembelajaran oleh sistem.

## 6.2 Pengujian dan Analisis Terhadap Pengaruh Nilai K

Pengujian dilakukan untuk mengetahui apakah perubahan nilai K dapat mempengaruhi nilai akurasi yang didapat. Pengujian dilakukan dengan mengubah nilai K dengan nilai 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 dan 15. Setiap nilai K dilakukan pengujian terhadap data uji dengan nilai E yang tetap, yaitu nilai E sebesar 3. Jumlah data latih yang digunakan ialah jumlah data latih terbaik yang dilakukan pada pengujian sebelumnya, yaitu jumlah data latih dengan persentase sebanyak 80% dari dataset. Jumlah data yang digunakan pada kelas *non-cancer* sebanyak 311 data, kelas kanker payudara sebanyak 96 data, kelas kanker usus sebanyak 97 data, dan kelas kanker paru-paru sebanyak 98 data. Hasil pengujian pengaruh nilai K ditampilkan pada Tabel 6.2 dan grafik hasil pengujian ditampilkan pada Gambar 6.3.

**Tabel 6.3 Hasil pengujian pengaruh nilai K**

| Persentase Data Latih | Persentase Data Uji | Nilai E | Nilai K | Akurasi Data Latih (%) |
|---|---|---|---|---|
| 80 (602 data) | 20 (150 data) | 3 | 2 | 66.000 |
| | | | 3 | 67.333 |
| | | | 4 | 67.333 |
| | | | 5 | 52.666 |
| | | | 6 | 54.000 |
| | | | 7 | 79.333 |
| | | | 8 | 80.666 |
| | | | 9 | 77.333 |
| | | | 10 | 72.000 |
| | | | 11 | 71.333 |
| | | | 12 | 64.000 |
| | | | 13 | 63.333 |
| | | | 14 | 62.666 |
| | | | 15 | 60.000 |




**Gambar 6.2 Grafik hasil pengujian pengaruh nilai K**

Berdasarkan Tabel 6.2 dan Gambar 6.2 dapat dilihat bahwa perubahan nilai K mempengaruhi nilai akurasi yang didapat. Hasil pengujian menunjukkan bahwa nilai akurasi yang didapatkan cukup beragam dan cenderung tidak stabil. Nilai akurasi berada pada nilai terbaik saat dilakukan pengujian dengan nilai K=8, setelah itu saat nilai K semakin besar, nilai akurasi yang dihasilkan akan semakin menurun. Penurunan nilai akurasi ini terjadi karena semakin besar nilai K yang

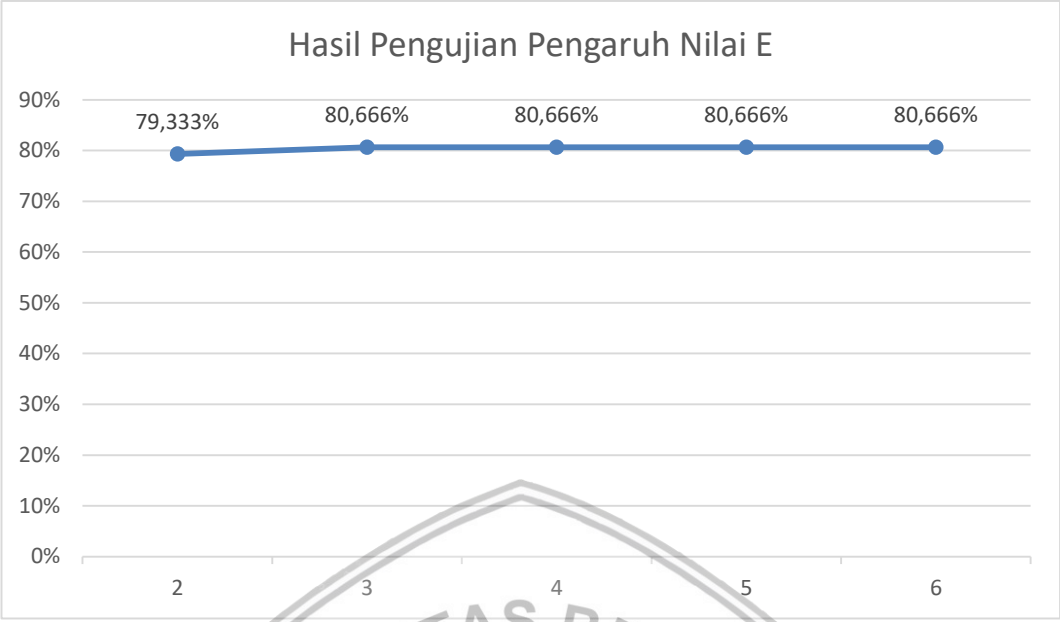
digunakan, semakin banyak pula data yang mempunyai jumlah jenis yang mendominasi masuk ketetanggaan yang menyebabkan saat dilakukan klasifikasi, data cenderung diklasifikasikan ke kelas yang salah. Terlebih dengan data yang tidak merata, jenis data yang memiliki jumlah yang mendominasi akan sering masuk pada ketetanggaan.

## 6.3 Pengujian dan Analisis Terhadap Pengaruh Nilai E

Pengujian dilakukan untuk mengetahui apakah perubahan nilai E dapat mempengaruhi nilai akurasi yang didapat. Pengujian dilakukan dengan mengubah nilai E dengan 2, 3, 4, 5 dan 6. Nilai persentase data latih yang digunakan adalah nilai persentase yang menghasilkan nilai akurasi yang baik pada pengujian sebelumnya, yaitu sebanyak 80% dari dataset. Jumlah data yang digunakan pada kelas non-cancer sebanyak 311 data, kelas kanker payudara sebanyak 96 data, kelas kanker usus sebanyak 97 data, dan kelas kanker paru-paru sebanyak 98 data. Selama perubahan nilai E dilakukan pada pengujian, nilai K yang digunakan adalah tetap. Nilai K yang digunakan adalah nilai K yang menghasilkan akurasi terbesar pada pengujian sebelumnya, yaitu nilai E sebesar 8. Hasil pengujian pengaruh nilai E ditampilkan pada Tabel 6.4 dan grafik hasil pengujian ditampilkan pada Gambar 6.3.

**Tabel 6.4 Hasil pengujian pengaruh nilai E**

| Persentase Data Latih (%) | Persentase Data Uji (%) | Nilai K | Nilai E | Akurasi Data Latih (%) |
|---|---|---|---|---|
| 80 (602 data) | 20 (150 data) | 8 | 2 | 79.333 |
| | | | 3 | 80.666 |
| | | | 4 | 80.666 |
| | | | 5 | 80.666 |
| | | | 6 | 80.666 |




**Gambar 6.3 Grafik hasil pengujian pengaruh nilai E**

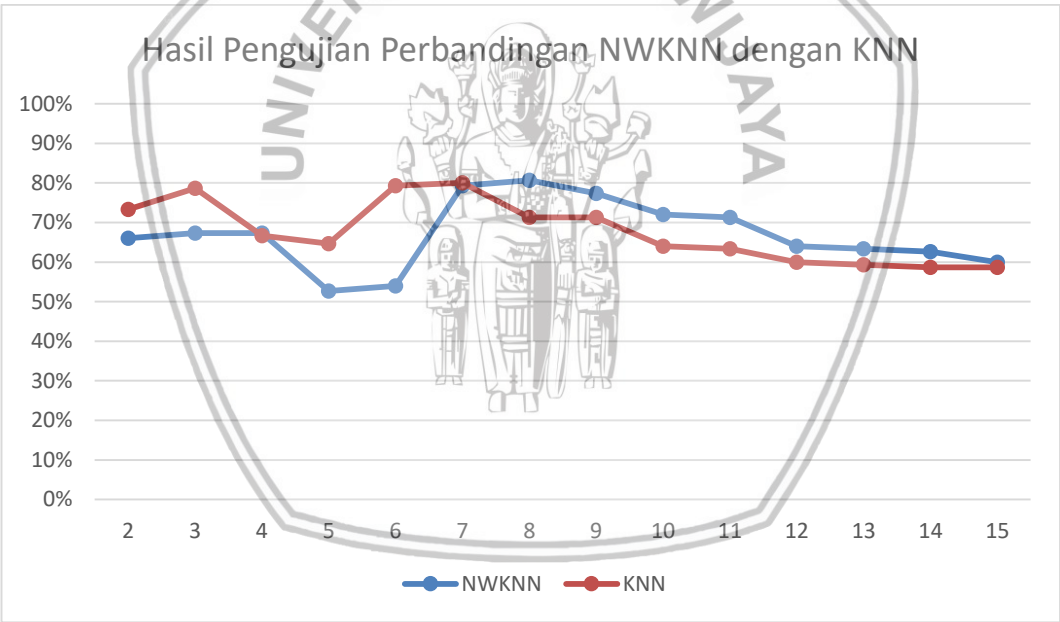
Berdasarkan Tabel 6.3 dan Gambar 6.3 dapat dilihat bahwa pengujian dengan mengganti nilai E cenderung memberikan nilai akurasi stabil. Hal ini menunjukkan bahwa perubahan pada nilai E tidak memberikan pengaruh besar yang berarti terhadap nilai akurasi yang didapat pada klasifikasi jenis kanker berdasarkan struktur protein. Nilai E ini memberikan pengaruh pada bobot kelas yang digunakan. Pada saat nilai E lebih besar dari 3, nilai bobot kelas yang memiliki jumlah mayoritas menjadi bobot yang tidak jauh beda dengan bobot kelas lainnya. Hal inilah yang menyebabkan nilai E tidak begitu berpengaruh terhadap akurasi NWKNN.

## 6.4 Pengujian Perbandingan Metode NWKNN dengan KNN

Pengujian dilakukan untuk mengetahui tingkat akurasi antara metode NWKNN dengan KNN dan mengetahui metode manakah yang menghasilkan akurasi terbaik. Pengujian dilakukan dengan mengubah nilai K dari nilai 2 hingga 15 pada tiap masing-masing pengujian. Rasio perbandingan data latih dengan data uji yang digunakan ialah 80%:20% dari dataset. Rasio perbandingan data latih dan data uji yang digunakan diambil dari pengujian sebelumnya yang menghasilkan akurasi tertinggi. Jumlah data yang digunakan pada kelas *non-cancer* sebanyak 311 data, kelas kanker payudara sebanyak 96 data, kelas kanker usus sebanyak 97 data, dan kelas kanker paru-paru sebanyak 98 data. Untuk pengujian NWKNN, nilai E yang digunakan ialah sebesar 3. Hasil pengujian perbandingan metode NWKNN dengan KNN ditampilkan pada Tabel 6.5 dan grafik hasil pengujian ditampilkan pada Gambar 6.4.

**Tabel 6.5 Hasil pengujian perbandingan metode NWKNN dengan KNN**

| Nilai K | NWKNN | KNN |
|---|---|---|
| 2 | 66.000 | 73.333 |
| 3 | 67.333 | 78.666 |
| 4 | 67.333 | 66.666 |
| 5 | 52.666 | 64.666 |
| 6 | 54.000 | 79.333 |
| 7 | 79.333 | 80.000 |
| 8 | 80.666 | 71.333 |
| 9 | 77.333 | 71.333 |
| 10 | 72.000 | 64.000 |
| 11 | 71.333 | 63.333 |
| 12 | 64.000 | 60.000 |
| 13 | 63.333 | 59.333 |
| 14 | 62.666 | 58.666 |
| 15 | 60.000 | 58.666 |
| Rata-Rata | 67.000 | 67.809 |




**Gambar 6.4 Grafik hasil pengujian perbandingan metode NWKNN dengan KNN**

Berdasarkan Tabel 6.5 dan Gambar 6.4, metode NWKNN menghasilkan akurasi tertinggi dengan akurasi sebesar 80.666% dengan menggunakan K sebesar 8, sedangkan metode KNN menghasilkan akurasi tertinggi dengan akurasi sebesar 80.000% dengan menggunakan K sebesar 7. Hal ini menandakan bahwa NWKNN menghasilkan akurasi sedikit lebih baik dibandingkan dengan metode KNN. Walaupun demikian, secara umum metode KNN memiliki performa yang lebih stabil dibanding NWKNN berdasarkan akurasi rata-rata yang dihasilkan dari keduanya, terlebih metode NWKNN mencapai akurasi terendah yang cukup drastis. Hal ini disebabkan adanya pengaruh pembobotan pada masing-masing kelas pada penghitungan NWKNN yang menyebabkan beberapa kelas terklasifikasi

dengan tidak tepat. Contohnya pada percobaan K=6, pada metode NWKNN sistem cenderung mengklasifikasikan data menjadi kelas kanker usus pada saat kelas sebenarnya adalah *non-cancer.* Hal ini disebabkan nilai K untuk ketetanggaan yang kurang luas untuk mencakup data kelas *non-cancer* dan pembobotan kelas menyebabkan skor menjadi lebih besar pada kelas minoritas seperti kelas kanker usus.

# BAB 7 PENUTUP

Bab ini membahas tentang kesimpulan yang dapat diambil terkait pada penelitian yang dilakukan dan juga saran untuk penelitian selanjutnya jika dilakukan penelitian yang serupa.

## 7.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis terhadap Klasifikasi Jenis Kanker Berdasarkan Struktur Protein dengan Menggunakan Metode *Neighbor Weighted K-Nearest Neighbor (NWKNN)*, dapat disimpulkan bahwa:

1. Klasifikasi jenis kanker berdasarkan struktur protein dilakukan dengan cara mengubah masing-masing data sekuens menjadi angka berdasarkan tabel matriks PAM. Setelah diubah menjadi angka, ditentukan nilai *similarity*nya menggunakan persamaan *Cosine Similiarity*. Setelah mendapatkan nilai *CosSim*, nilai masing-masing data diurutkan *CosSim*nya dari yang terbesar hingga yang terkecil. Kemudian diambil beberapa data dari yang terbesar sebanyak nilai K untuk dijadikan ketetanggaan. Selanjutnya dihitung pembobotannya agar bisa didapatkan nilai skor masing-masing kelas. Kelas yang memiliki nilai skor tertinggi akan menjadi kelas hasil klasifikasi.
2. Pada pengujian parameter klasifikasi jenis kanker menggunakan metode NWKNN, perubahan nilai K yang digunakan bernilai dari 2 hingga 15 dan nilai E yang digunakan bernilai dari 2 hingga 6. Berdasarkan hasil pengujian, metode NWKNN dapat melakukan klasifikasi jenis kanker berdasarkan struktur protein dengan baik karena sistem menghasilkan akurasi terbaik sebesar 80.666% pada K=8 dan E=3 dengan data latih sebanyak 80% dari dataset (602 data) dan data uji sebanyak 20% (150 data).
3. Metode NWKNN menghasilkan akurasi yang lebih baik dibandingkan dengan metode KNN untuk mengklasifikasi jenis kanker berdasarkan struktur protein. Metode NWKNN menghasilkan akurasi tertinggi sebesar 80.666% sedangkan metode KNN menghasilkan akurasi tertinggi sebesar 80.000%.

## 7.2 Saran

Saran yang dapat diberikan pada penelitian klasifikasi jenis kanker berdasarkan struktur protein menggunakan metode NWKNN adalah:

1. Penelitian selanjutnya diharapkan dapat mengembangkan sistem dengan metode yang berbeda atau mengombinasikan metode NWKNN dengan metode lain agar dapat memperoleh hasil akurasi yang lebih baik.
2. Pada penelitian ini data hanya menggunakan satu fitur, yaitu fitur berupa struktur protein saja untuk mengklasifikasi jenis kanker. Untuk penelitian selanjutnya bisa menggunakan data dengan tambahan fitur lainnya guna membantu meningkatkan akurasi sistem dalam mengklasifikasi jenis kanker.

# DAFTAR PUSTAKA


Anggorowati, L., 2013. Faktor Risiko Kanker Payudara Wanita. *Jurnal Kesehatan Masyarakat,* 8(2), pp. 121-126.

Anggraeni, C. D. R., 2017. *Jumlah Penderita Kanker Di Indonesia Tiap Tahun Makin Meningkat.* [Online] Available at: http://artikel.allianz.co.id/agen/detail-article/Jumlah-Penderita-Kanker-Di-Indonesia-3817 [Diakses 14 Februari 2018].

Biro Komunikasi dan Pelayanan Masyarakat, K. K. R., 2017. *Kementerian Kesehatan Ajak Masyarakat Cegah Dan Kendalikan Kanker.* [Online] Available at: http://www.depkes.go.id/article/print/17020200002/kementerian-kesehatan-ajak-masyarakat-cegah-dan-kendalikan-kanker.html [Diakses 14 Februari 2018].

Desideria, B., 2017. *Kasus Kanker Usus Besar di Indonesia Meningkat.* [Online] Available at: http://health.liputan6.com/read/2887429/kasus-kanker-usus-besar-di-indonesia-meningkat [Diakses 15 Februari 2018].

Fadila, P. N., 2016. *Identifikasi Jenis Attention Deficit Hyperactivity Disorder (ADHD) Pada Anak Usia Dini Menggunakan Metode Neighbor Weighted K-Nearest Neighbor (NWKNN).* Malang: Universitas Brawijaya.

Harahap, I. A., 2004. *Perawatan Pasien Dengan Kolostomi Pada Penderita Cancer Colorectal.* Medan: Universitas Sumatera Utara.

Hermawati, F. A., 2013. *Data Mining.* Yogyakarta: Penerbit Andi.

Jones, N. C. & Pevzner, P. A., 2004. *An Introduction to Bioinformatics Algorithms.* Massachusetts London: The MIT Press Cambridge.

Katili, A. S., 2009. Struktur Dan Fungsi Protein Kolagen. *Jurnal Pelangi Ilmu,* 2(5), pp. 19-29.

Kementerian Kesehatan RI, 2015. *InforDATIN Pusat Data dan Informasi Kementerian Kesehatan RI.* [Online] Available at: http://www.depkes.go.id/resources/download/pusdatin/infodatin/infodatin-kanker.pdf [Diakses 17 November 2018].

Kurnianti, R., 2013. *Penggunaan Metode Pengelompokkan K-Means Pada Klasifikasi KNN Untuk Penentuan Jenis Kanker Berdasartkan Susunan Protein.* Malang: Universitas Brawijaya.

Meristika, Y. S., 2013. *Perbandingan K-Nearest Neighbor dan Fuzzy K-Nearest Neighbor Pada Diagnosis Penyakit Diabetes Melitus.* Malang: Universitas Brawijaya.

Mulyana, S., 2013. *Penerapan Hidden Markov Model Dalam Clustering Sequence Protein Globin.* Malang: Universitas Brawijaya.

Prasetyo, E., 2012. *Data Mining − Konsep dan Aplikasi Menggunakan MATLAB.* Yogyakarta: Penerbit Andi.

Pusat Data dan Informasi Kementerian Kesehatan RI, 2015. *Situasi Penyakit Kanker.* [Online]
Available at: http://www.depkes.go.id/resources/download/pusdatin/buletin/buletin-kanker-old.pdf
[Diakses 17 November 2018].

Retwitasari, A., 2016. *Penentuan Jenis Kanker Berdasarkan Struktur Protein Menggunakan Algoritma Modified K-Nearesrt Neighbor (MKNN).* Malang: Universitas Brawijaya.

Rivaldi, A., 2017. *Klasifikasi Penyimpangan Tumbuh Kembang Pada Anak Menggunakan Metode Neighbor Weighted K-Nearest Neighbor (NWKNN).* Malang: Universitas Brawijaya.

Rizby, L. P., 2018. *Clustering Pasien Kanker Berdasarkan Struktur Protein Dalam Tubuh Menggunakan Metode K-Medoids.* Malang: Fakultas Ilmu Komputer Universitas Brawijaya.

Sari, M. I., 2007. *Struktur Protein.* Medan: Universitas Sumatera Utara.

Sudhakar, A., 2009. History of Cancer, Ancient and Modern Treatment Methods. *Journal of Cancer Science & Therapy,* 1(2), pp. i-iv.

Suryanis, A., 2017. *Kanker Penyebab Kematian Ke-3 Penyakit Tidak Menular.* [Online]
Available at: https://gaya.tempo.co/read/873068/kanker-penyebab-kematian-ke-3-penyakit-tidak-menular
[Diakses 13 Februari 2018].

Tan, S., 2005. Neighbor-Weighted K-Nearest Neighbor For Unbalanced Text Corpus. *Expert Systems with Applications,* Issue 28, pp. 667-671.

Utami, T. N., 2018. *Implementasi Fuzzy k-Nearest Neighbor (Fk-NN) untuk Klasifikasi Jenis Kanker berdasarkan Susunan Protein.* Malang: Universitas Brawijaya.

Viswasnathan, V., 2016. *Hubungan Antara Merokok Dengan Terjadinya Kanker Paru di Departemen Pulmonologi FKUSU/RSUP H.Adam Malik Medan Tahun 2014.* Medan: Universitas Sumatera Utara.

Wulandari, T., 2018. *Klasifikasi Jenis Kanker Berdasarkan Struktur Protein Menggunakan Algoritma Naive Bayes.* Malang: Fakultas Ilmu Komputer Universitas Brawijaya.

Yulianti, I., Setyawan, H. & Sutiningsih, D., 2016. Faktor-Faktor Risiko Kanker Payudara (Studi Kasus Pada Rumah Sakit Ken Saras Semarang). *JURNAL KESEHATAN MASYARAKAT (e-Journal),* 4(4), pp. 401-409.